INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 779

METHODES IMPLICITES EFFICACES POUR LA RESOLUTION DES EQUATIONS D'EULER EN ELEMENTS FINIS

Hervé STEVE

DECEMBRE 1987

MÉTHODES IMPLICITES EFFICACES POUR LA RÉSOLUTION DES ÉQUATIONS D'EULER EN ÉLÉMENTS FINIS

EFFICIENT IMPLICIT SOLVERS FOR EULER EQUATIONS WITH A FINITE ELEMENT METHOD

Hervé STEVE

INRIA Sophia-Antipolis
2004, Route des Lucioles
Parc de Sophia-Antipolis 1 et 2
06560 VALBONNE

METHODES IMPLICITES EFFICACES DE RESOLUTION DES EQUATIONS D'EULER EN ELEMENTS FINIS

Résumé :

Cette étude est consacrée à la résolution numérique des équations d'Euler stationnaires en éléments finis à l'aide de schémas implicites linéarisés décentrés d'ordre élevé en espace. Ces schémas possèdent de bonnes propriétés : utilisation possible de grands pas de temps, solution numérique sans oscillation en ordre un tandis qu'un procédé de limitation a été utilisé en ordre deux pour préserver la monotonie , les calculs ne nécessitent pas de réglage de viscosité artificielle. De plus on peut augmenter l'efficacité de ces schémas par l'utilisation de grands CFL progressivement jusqu'à $\simeq 10^{17}$.

L'application de ces schémas dans un contexte industriel pose des problèmes sérieux d'encombrement mémoire lorsque la matrice provenant de la discrétisation des termes implicites est complètement stockée. Nous proposons donc dans ce rapport plusieurs méthodes dites à faible stockage matriciel. Enfin, nous terminons ce rapport en décrivant l'implémentation des différents codes proposés sur machine vectorielle.

EFFICIENT IMPLICIT SOLVERS FOR EULER EQUATIONS WITH A FINITE ELEMENT METHOD

Abstract :

This study is devoted to the numerical solution of the steady Euler equations using a finite element method by several high-order upwind (linearized -) implicit schemes. These schemes have several advantageous features : large time-steps can be used, finite-element-type unstructured meshes can be employed, artificial viscosity need not be added and the first order accurate scheme is oscillation-free, while, in the second-order scheme some limiter can be used to preserve monotonicity. Efficiency is achieved via the implicit formulation. In some cases CFL numbers of the order of 10^{17} have been used.

However, the extension of these schemes to the industrial context sets great problems related to memory-storage requirements since, the solution procedure beeing implicit, a large matrix should normally be stored. Then the report proposes several low-storage methods. Finally, this work is achieved by describing the adaptation to a vector computer.

Table des Matières

I. Introduction

On s'intéresse dans cette étude à la résolution numérique des équations d'Euler bidimensionnelles pour des fluides parfaits compressibles en régime transsonique et supersonique y compris pour de grands nombres de Mach (plusieurs dizaines). Notons qu'à partir d'un nombre de Mach supérieur à cinq, le modèle des équations d'Euler est insuffisant à cause de l'apparition de réactions chimiques (écoulements réactifs); dans cette étude les phénomènes chimiques ne seront pas pris en compte.

On se restreindra au calcul des solutions stationnaires obtenues par une approche instationnaire reposant sur des schémas numériques permettant de grands pas de temps. Il en résulte la mise au point de schémas plus complexes mais plus efficaces que les schémas explicites usuels (schémas d'Euler par exemple). C'est ce que nous montrerons à travers une étude de stabilité d'une part (Partie I), et, par des expériences numériques d'autre part.

Un certain nombre de méthodes numériques permettent des grands pas de temps; nous citerons parmi ces méthodes celles de type implicite :

1) Dans le cas des maillages structurés :

– Les schémas implicites centrés :
R. Beam et R.F. Warming [25] ont été parmi les premiers à les introduire dans le cadre des différences finis finies. On trouvera une étude détaillée de ces schémas dans la thèse de A. Lerat [16] qui présente par ailleurs une version avec lissage de résidu. Toutes ces méthodes aboutissent à des systèmes linéaires tridiagonaux par blocs en 1-D, et peuvent s'étendre aisément au cas bidimensionnel par des techniques de résolution A.D.I.

– Les schémas implicites décentrés. Nous citerons parmi d'autres :
Le schéma proposé par Mc Cormack [29] où le système est bidiagonal en 1-D.
W.A. Mulder et B. van Leer [23] construisent un schéma implicite linéarisé non factorisé.

– Les méthodes implicites multigrilles : P. W. Hemker et S. P. Spekreijse [28].

2) Dans le cas des maillages non structurés :

– Le schéma implicite centré de A. Lerat [16] a été étendu aux éléments finis (voir F. Angrand, etc [8]).

– En décentré, l'adaptation de schémas implicites linéarisés au cas des maillages

non structurés (éléments finis) a été étudiée par B. Stoufflet [2]. Les schémas sont construits à partir d'une linéarisation du flux décentré de Vijayasundaram [5], pour obtenir une version implicite d'un schéma centré ou décentré.

On trouvera une extension du schéma précédent à d'autres flux décentrés dans [31]. Les systèmes linéaires (tridiagonal par blocs en 1-D) y sont résolus par des méthodes de relaxation de type Gauss-Seidel. De grands pas de temps on pu être utilisés (C.F.L. de l'odre de 10^3) pour le calcul d'écoulements transsoniques [2]. La robustesse de ce schéma a été testée numériquement par des calculs plus complexes autour d'un engin spatial en forte incidence et à grand nombre de Mach (en 3-D) [10].

Les premières tentatives d'utilisation de cette méthodes sur des maillages fins ont mis en évidence la principale faiblesse de la méthode : le stockage prohibitif des termes implicites. Nous proposons ici une solution à ce problème d'encombrement mémoire. Deux algorithmes avec stockage partiel de la matrice implicite sont étudiés. Dans le premier algorithme proposé par A. Eberle [13], seuls les termes diagonaux sont calculés et utilisés. Nous introduisons un autre schéma dans lequel la partie diagonale est stockée et les termes extra-diagonaux sont calculés à chaque balayage de la résolution du système linéaire. Ce schéma conserve les propriétés de consistance, de stabilité et de précision du schéma initial avec stockage complet. Nous comparons l'efficacité de ces deux méthodes dans le contexte des éléments finis.

Le travail que nous présentons se décompose en trois parties :

Dans la première partie on étudiera la convergence du schéma implicite linéarisé décentré avec une approximation spatiale d'ordre un ou d'ordre deux : on établit le lien avec la méthode de Newton dans le cas stationnaire. De plus une analyse de stabilité est développée dans le cas scalaire linéaire. On trouvera des études similaires dans les travaux de B. van Leer et W. Mulder [23], de C. Jespersen et T. Pulliam [18], de B. Stoufflet [2] et de J-A. Désidéri [9].

La seconde partie est consacrée au problème du stockage des matrices : présentation des méthodes **sans-stockage** c'est à dire à faible encombrement mémoire.

Enfin, la troisième partie traitera de l'adaptation de ces algorithmes au calcul vectoriel : *coloriage* de la géométrie, choix de la méthode de résolution linéaire, ... etc. On trouvera une présentation des différentes techniques utilisées ici pour le calcul vectoriel en éléments finis dans [14].

II. Méthodes implicites à convergence rapide

1. Equations d'Euler

Lois de conservation :

Le système des équations d'Euler à deux dimensions d'espace s'écrit en formulation conservative :

$$\begin{cases} \dfrac{\partial}{\partial t}W(x,y,t) + \dfrac{\partial}{\partial x}F(W(x,y,t)) + \dfrac{\partial}{\partial y}G(W(x,y,t)) = 0 & \text{sur } \Omega \times I\!R^{+*} \\ W(x,y,0) = W_0(x,y) & \text{sur } \Omega \times \{0\} \\ W(x,y,t) = W_\infty(x,y,t) & \text{sur } \overline{\Omega} \times I\!R^{+*} \end{cases} \tag{1}$$

Où Ω est un ouvert borné de $I\!R^2$ de frontière $\partial\Omega = \Gamma$, $W(x,y,t)$ est une fonction vectorielle de $I\!R^4$ définie sur $\overline{\Omega} \times I\!R^+$.

Par la suite nous noterons les dérivées partielles de type $\dfrac{\partial}{\partial x}W(x)$ par W_x.

Le système (1) s'écrit dans ces notations :

$$\begin{cases} W_t + F(W)_x + G(W)_y = 0 & \text{sur } \Omega \times I\!R^{+*} \\ + & \text{conditions initiales et conditions aux bords} \end{cases} \tag{2}$$

Les fonctions de flux $F(W)$ et $G(W)$ définies dans $I\!R^4$ sont :

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} \quad ; \quad F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{pmatrix} \quad ; \quad G(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{pmatrix}$$

Où ρ est la masse volumique, (u,v) sont les composantes de la vitesse du fluide, E représente l'énergie totale par unité de volume, et, p est la pression, vérifiant la *loi d'état des gaz parfaits*, $p = (\gamma - 1)(E - \dfrac{1}{2}\rho(u^2+v^2))$ où γ est le rapport des chaleurs spécifiques supposé constant ici : $\gamma = \dfrac{7}{5} = 1,4$ pour un gaz parfait diatomique.

Forme quasi-linéaire et hyperbolicité :

Les fonctions F et G sont des fonctions homogènes de degré un, cette propriété sera utilisée dans le §2. On peut écrire (2) sous la forme d'un système quasi-linéaire:

$$\begin{cases} W_t + A(W)W_x + B(W)W_y = 0 & \text{sur } \Omega \\ + & \text{conditions initiales et aux limites} \end{cases}$$

où les matrices A, B de $\mathbb{R}^4 \times \mathbb{R}^4$ sont les dérivées ou jacobiens des fonctions de flux d'Euler : $A(W) = F'(W)$; $B(W) = G'(W)$.

Considérons alors une combinaison linéaire de flux :
$\mathcal{F}_\mu(W) = \mu^x F(W) + \mu^y G(W)$ où $\vec{\mu} = (\mu^x, \mu^y)$ est un vecteur non nul quelconque de $\mathbb{R}^2$.

On rappelle que le système quasi-linéaire écrit avec ce nouveau flux $\mathcal{F}$ est hyperbolique c'est à dire que le jacobien : $\mathcal{A}_\mu(W) = \mu^x A(W) + \mu^y B(W)$ est diagonalisable et que toutes ses valeurs propres sont réelles pour tout $W \in \mathbb{R}^4$ et $\vec{\mu} \in \mathbb{R}^2$. Les valeurs propres notées $\lambda^k_\mu(W)$ $(k = 1, ..4)$ sont :

$$\begin{cases} \lambda^1_\mu(W) = \begin{pmatrix} u \\ v \end{pmatrix} . \vec{\mu} + c \\ \lambda^2_\mu(W) = \lambda^3_\mu(W) = \begin{pmatrix} u \\ v \end{pmatrix} . \vec{\mu} \\ \lambda^4_\mu(W) = \begin{pmatrix} u \\ v \end{pmatrix} . \vec{\mu} - c \end{cases} \tag{3}$$

Où c est la vitesse locale du son qui vaut : $c = \sqrt{\dfrac{\gamma p}{\rho}}$.

Nous diagonalisons la matrice $\mathcal{A}_\mu$ que nous notons $\mathcal{A}$: $\mathcal{A} = T^{-1}\Lambda\, T$ avec $\Lambda = diag.(\lambda^k)$ matrice des valeurs propres λ^k de $\mathcal{A}$, et T matrice de transformation inversible.

Par la suite nous utiliserons la décomposition suivante :

$$\mathcal{A} = \mathcal{A}^+ + \mathcal{A}^- \ ; \ |\mathcal{A}| = \mathcal{A}^+ - \mathcal{A}^-$$

De même, $\mathcal{A}^+$ et $\mathcal{A}^-$ sont des matrices diagonalisables avec la même transformation T, $\mathcal{A}^\pm = T^{-1}\Lambda^\pm\, T$ où :

$$\Lambda^+ = diag.(\max(\lambda^k, 0)) \ ; \ \Lambda^- = diag.(\min(\lambda^k, 0))$$

Nous introduisons le nombre de Mach local M : $M = \frac{\sqrt{u^2+v^2}}{c}$. Si $M < 1$, nous dirons que l'écoulement est localement *subsonique* ; et si $M \geq 1$ l'écoulement sera localement *supersonique.*

Conditions aux bords :
Pour un écoulement externe, nous distinguons la frontière *à l'infini* notée Γ_∞ et la frontière sur le bord de l'obstacle (paroi) notée Γ_B. (voir figure **I.**):

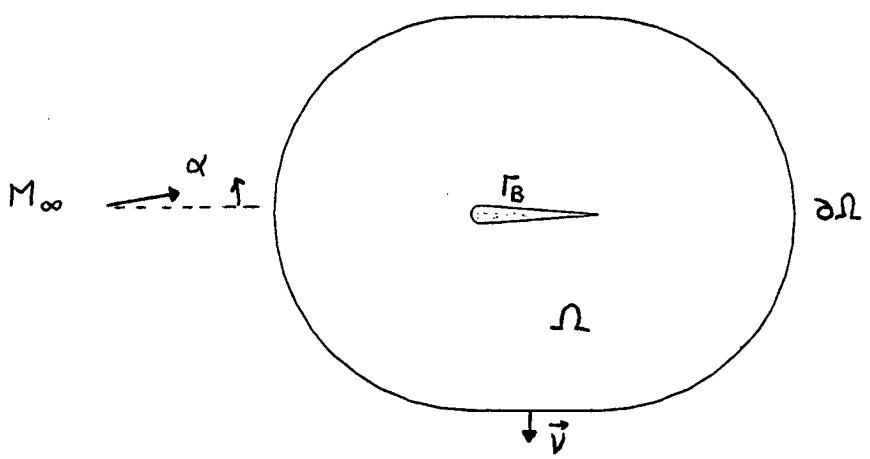


Fig. I : *Domaine de calcul pour un écoulement externe*

Sur Γ_∞ nous supposerons que l'écoulement est uniforme. Nous calculons W_∞ avec :

$$\rho_\infty = 1 \; ; \; \vec{V}_\infty = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \; ; \; p_\infty = \frac{\rho_\infty}{\gamma M_\infty^2}$$

Où α est l'angle d'attaque de l'obstacle par rapport au fluide et M_∞ est le nombre de Mach à l'infini.

Sur Γ_B nous imposons la condition de glissement : $\vec{V} \cdot \vec{\nu} = 0$ avec $\vec{\nu}$ vecteur normal.

Conditions initiales :
Nous étendons à tout le domaine les conditions imposées à l'infini : $W(x,y,t=0) = W_0(x,y) = W_\infty$ sur tout le domaine Ω.

Existence et unicité d'une solution stationnaire

Dans cette étude, nous recherchons des solutions stationnaires du problème instationnaire (1). Le problème stationnaire s'écrit :

$$\begin{cases} F(W)_x + G(W)_y = 0 & \text{sur } \Omega \\ + & \text{conditions aux bords} \end{cases} \tag{4}$$

Le système (4) n'est pas résolu directement : nous résolvons le problème évolutif (2) pour obtenir une solution physiquement acceptable.

Rappelons que le problème non linéaire instationnaire n'a pas toujours de solution classique définie pour pour tout $t > 0$ même si la condition initiale est très régulière, en effet le caractère hyperbolique des équations conduit à de solutions discontinues. P.D. Lax [34] introduit alors la notion de solution faible, qui vérifie :

$$\iiint_{\Omega \times]0,\infty[} (\varphi_t W + \varphi_x F(W) + \varphi_y G(W))\, dx\, dy\, dt \quad + \quad \iint_{\Omega} \varphi(x,y,0) W_0(x,y) dx\, dy = 0$$

Où φ est une fonction définie dans $\mathbb{R}^m$, continûment différentiable et à support compact dans $\mathbb{R}^2 \times [0,\infty[$. Dans le cas scalaire ($m = 1$), un théorème d'unicité de la solution faible au sens défini ci-dessus vérifiant une condition d'entropie a éte démontré (S.N. Kruzhkov [32]), l'existence de cette solution est obtenue par des méthodes de viscosité s'appuyant essentiellement sur la vérification du principe du maximum et sur la donnée d'une estimation de la variation bornée (caractère de monotonie). Pour une bibliographie détaillée nous renvoyons à [16]. Mais dans le cas général des systèmes (équations d'Euler par exemple) le problème reste ouvert. Dans le cas discret, des études analogues ont été menées pour définir et déterminer si un schéma numérique est entropique c'est à dire s'il y a convergence vers une solution vérifiant une condition d'entropie. On citera par exemple les travaux de Leroux [33] sur les schémas de type Godunov, ceux d'Osher sur les E-schémas [7], ... etc.

2. Schéma implicite linéarisé

2.1. *Méthode de Newton*

Pour des raisons de simplicité nous considérons les équations d'Euler dans le cas monodimensionnel. Nous rappelons le système à résoudre analogue à (1) :

$$\begin{cases} W_t + F(W)_x = 0 \\ \qquad\qquad + \text{conditions initiales sur } \Omega \subset \mathrm{R} \\ \qquad\qquad + \text{conditions aux bords sur } \partial\Omega \end{cases}$$

Avec $W, F(W)$ vecteurs de $\mathbb{R}^m$ ($m = 3$ pour les équations d'Euler).

L'équation instationnaire associée au système (5) peut s'écire de façon quasi-linéaire :

$$W_t + A(W)W_x = 0 \tag{5}$$

avec $A(W)$ matrice jacobienne de $F(W)$ dans $\mathbb{R}^m \times \mathbb{R}^m$.

Le schéma totalement implicite s'écrit après discrétisation temporelle :

$$\frac{W^{n+1} - W^n}{\Delta t^n} + G(W^{n+1}) = 0 \tag{6}$$

où Δt^n désigne le pas de temps variable entre deux instants consécutifs n et $n+1$: $\Delta t^n = t^{n+1} - t^n$, $W^n = W(x, y, t^n)$; et, où $G(W)$ approche le terme $\{F(W)_x\}$.

La linéarisation du flux à l'aide d'un développement de Taylor à l'ordre un en temps lorsque G(W) est une fonction différentiable s'exprime :

$$G(W^{n+1}) = G(W^n) + G'(W^n)(W^{n+1} - W^n) + o(\Delta t^2)$$

Avec $G'(W^n)$ jacobien de $G(W^n)$.

D'où la version linéarisé du schéma implicite [25]:

$$\left(\frac{I}{\Delta t^n} + G'(W^n)\right) \delta W^{n+1} = -G(W^n) \tag{7}$$

Où on note $\delta W^{n+1} = W^{n+1} - W^n$ et I la matrice identité de $\mathbb{R}^m \times \mathbb{R}^m$.
Si le pas de temps est infiniment grand, l'algorithme (7) devient une méthode de Newton, qui a une convergence quadratique au voisinage de la solution du problème $G(W) = 0$:

$$G'(W^n)\delta W^{n+1} = -G(W^n)$$

La précision spatiale de cette méthode appliquée à la résolution stationnaire de (5) est donnée par l'ordre de précision du calcul du flux $G(W)$.

Lorsque le flux $G(W)$ n'est pas différentiable ou lorsque la différenciation du flux est complexe voir coûteuse, on introduit un opérateur linéaire que nous notons P^n qui *approche* le jacobien G'. Le sytème (7) s'écrit alors :

$$\left(\frac{I}{\Delta t^n} + P^n\right) \delta W^{n+1} = -G(W^n) \tag{8}$$

Comme P^n est différent du vrai jacobien $G'(W^n)$, le schéma (8) pour Δt grand n'est plus en général une méthode de Newton mais *une méthode itérative préconditionnée* ou de Newton modifiée :

$$P^n \delta W^{n+1} = -G(W^n) \tag{9}$$

2.2. *Calcul de flux et décentrage 1-D*

Définitions et notations :

On note x_i ; $i = 0,..N$ les nœuds du maillage et $\Delta x = x_{i+1} - x_i$ le pas d'espace. On pose en chaque point x_i : $W_i = W(x_i)$ constant sur chaque intervalle $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. On définit la fonction de flux $G(W)$ en chaque point x_i :

$$G(W)_i = \frac{1}{\Delta x}(\Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}})$$

Où Φ est le flux numérique qui s'écrit dans le cas du schéma à trois points du premier ordre en espace :

$$\Phi_{i+\frac{1}{2}} = \Phi(W_i, W_{i+1}) \quad ; \quad \Phi_{i-\frac{1}{2}} = \Phi(W_{i-1}, W_i)$$

Dans le cas des équations d'Euler, on peut définir de façon générale le flux Φ pour un schéma soit centré ou décentré (en espace) (cf. [35]) :

$$\Phi(U,V) = \frac{1}{2}\left(F(U) + F(V) - d(U,V)\right)$$

Avec U, V deux vecteurs de $\mathbb{R}^m$ qui sont les états à droite et à gauche.

La fonction $d(U,V)$ contient les termes de viscosité et donc précise le schéma utilisé. Nous dirons, suivant [35], qu'un schéma est décentré s'il peut s'écrire sous la forme :

$$d(U,V) = \mid A(\frac{U+V}{2}) \mid (V-U) \quad + \quad o(\mid V-U \mid)$$

La condition de consistance de Φ est réalisée par construction : $\Phi(U,U) = F(U)$ ($d(U,U) = 0$).

Nous utiliserons dans cette étude deux fonctions de flux numériques qui nous permettent d'obtenir des solutions sur de nombreux types d'écoulement :

a) La décomposition de flux de Steger-Warming [6] :

$$d^{SW}(U,V) = \mid A(V) \mid V - \mid A(U) \mid U \tag{10}$$

Le flux de Steger-Warming est totalement décentré et il s'écrit plus simplement:

$$\Phi^{SW}(U,V) = A^{+}(U)U + A^{-}(V)V$$

b) La décomposition du flux de Osher [7] :

$$d^{OSH}(U,V) = \int_{U}^{V} \mid A(W) \mid \, dW \tag{11}$$

D'où le flux :

$$\Phi^{OS}(U,V) = \frac{1}{2}\left(F(U) + F(V) - \int_{U}^{V} \mid A(W) \mid \, dW \right)$$

Où le chemin d'intégration (U,V) est défini sur les caractéristiques en utilisant la solution du problème de Riemann pour les équations d'Euler [7], [34].

Une description plus détaillée de ce flux est donnée dans la Partie IV.

Remarques :

Le calcul du jacobien du flux de Osher écrit à l'aide des variables conservatives $(\rho, \rho u, E)$ est coûteux et complexe. On trouvera dans [36] une écriture simplifiée de ce schéma avec d'autres variables $(c, u, log\frac{p}{\rho^{\gamma}})$.

Nous utilisons ici le flux de Osher pour le calcul des termes explicites (i.e. $G(W)$) et la décomposition de flux de Steger-Warming pour le calcul des termes implicites (i.e. $P(W)$). La robustesse de ce schéma implicite résultant (dit Steger-Warming/Osher) a pu être testée numériquement par des calculs d'écoulements tridimensionnels complexes [10].

δ-schéma :

Nous posons : $M^n = I + \Delta t^n P^n$, M^n est alors la matrice implicite. Dans le cas du schéma de Steger-Warming, M^n est tridiagonale par blocs ($m \times m$).
$M^n = Tridiag.(L^n, D^n, U^n)$ avec :

$$\begin{cases} L^n = -\sigma A^+(W^n_{i-1}) \\ D^n = I + \sigma(A^+(W^n_i) - A^-(W^n_i)) \\ U^n = \sigma A^-(W^n_{i-1}) \end{cases} \tag{12}$$

Pour $\sigma = \dfrac{\Delta t^n}{\Delta x}$.

Choix du pas de temps :

On ne peut pas calculer les solutions stationnaires des équations d'Euler avec un choix arbitraire du pas de temps en commencant le processus itératif par des grands pas de temps. Il semble que la condition initiale choisie (évidemment uniforme) n'étant pas en général proche de l'état stationnaire, nous éloigne des conditions de convergence de la méthode de Newton (convergence quadratique au voisinage de la solution). B. van Leer et W. Mulder [23] proposent une formule de pas de temps pour leur formulation SER (*Switch Evolution Relaxation*) tenant compte du résidu itéré :

$$\Delta t^n = \frac{K}{RES^n}$$

Où K est une constante qui dépend des données du problème (proche de l'unité en général) et RES^n est défini par :

$$RES^n = \frac{\|G(W^n)\|_2}{\|G(W^0)\|_2}$$

Où $\| \cdot \|_2$ désigne la norme L_2.

Algorithme implicite :

Nous donnons l'algotithme sous la forme d'un schéma à deux phases : une phase explicite qui prend en compte les données physiques du problème à résoudre :

1 : Phase physique / explicite , flux de Osher

$$\widehat{\delta W} = -\sigma G(W^n) \tag{13}$$

Puis une phase implicite qui met en œuvre les procédés mathématiques de résolution : .

2 : Phase mathématique / implicite , flux de Steger-Warming A chaque pas de temps on résoud le système linéaire suivant :

$$\begin{cases} M^n \delta W^{n+1} = \widehat{\delta W} \\ W^{n+1} = W^n + \delta W^{n+1} \end{cases} \tag{14}$$

Propriétés du schéma implicite :

Rappelons quelques propriétés relatives aux schémas décrits ci-dessus et qui sont démontrées dans la thèse de B. Stoufflet [2] :

Proposition 1 :

Le schéma de type (13)-(14) est précis au premier ordre en temps et en espace.

Proposition 2 :

Le schéma implicite linéarisé décentré ordre 1 est linéairement inconditionnellement stable dans le cas scalaire

Proposition 3 :

La matrice M^n est à diagonale dominante par blocs dans le cas scalaire. Elle est par ailleurs inversible (voir A. Lerat [16]).

2.3. *Résolution du système linéaire*

Bien que la Proposition 3 soit démontrée seulement dans le cas scalaire, nous utilisons des méthodes de relaxations, Jacobi et Gauss-Seidel, pour résoudre le système linéaire (14). La convergence de la phase de résolution linéaire peut nécessiter un grand nombre d'itérations, par exemple si le système est mal-conditionné. Nous étudions si les propriétés de consistance, de précision et de stabilité sont maintenues lorsqu'un nombre limite d'itérations linéaires est réalisé.

Le modèle scalaire 1-D de l'équation de propagation des ondes en périodique s'écrit :

$$\begin{cases} U_t + cU_x = 0 & \text{sur } \mathbb{R}^+ \times \mathbb{R} \\ U(x+2\pi,t) = U(x,t) & (\text{ périodicité en espace }) \end{cases} \tag{15}$$

Avec c réel non nul.

Soit le maillage 1-D défini au §2.2 : $x_i = 0,..,N+1$. Nous obtenons alors le δ-schéma implicite (voir (13)-14)) :

$$M^n \delta U_i^n = \widehat{\delta U_i}$$

Nous remarquons que dans le cas de l'équation scalaire (15) tous les schémas décentrés se réduisent à :

$$\begin{aligned} \widehat{\delta U_i} &= -\sigma(\Phi^n_{i+\frac{1}{2}} - \Phi^n_{1-\frac{1}{2}}) \\ &= -\sigma(c^+U_i^n + c^-U_{i+1}^n - c^+U_{i-1}^n - c^-U_i^n) \\ &= -\sigma(-c^+U_{i-1}^n + \mid c \mid U_i + c^-U_{i-1}^n) \end{aligned}$$

$$M^n = Tridiag.(\ -\sigma c^+\ ;\ 1+\sigma \mid c \mid\ ;\ \sigma c^-\ )$$

Avec : $c^+ = \dfrac{c+\mid c\mid}{2}$; $c^- = \dfrac{c-\mid c\mid}{2}$.

La méthode de Gauss-Seidel :

Une itération linéaire peut être obtenue de deux façons : elle est dite *croissante* si la boucle sur les points d'interpolation ou nœuds est faite dans le sens ascendant, le numéro de nœud i variant de 0 à $N+1$; au contraire elle est *décroissante* si la boucle est réalisée dans le sens descendant, le nœud i variant de $N+1$ à 0.

L'étude de stabilité du schéma (16) est faite ici au moyen de **l'analyse de Fourier**. Notons par U_j^n les modes de Fourier de la forme :

$$U_j^n = \exp(ij\xi_k)\ ;\ \xi_k = \frac{k2\pi}{N+1}\ \ (k = 1,..N)\ ;\ i = \sqrt{-1}$$

U_j^{n+1} s'écrit en fonction de U_j^n :

$$U_j^{n+1} = g_k(\Delta t)U_j^n$$

Où $g_k(\Delta t)$ est le facteur d'amplification du schéma considéré. Nous dirons que le schéma est stable si la condition de von Neumann est vérifée :

$$\max_{k\in[1;N]} \mid g_k(\Delta t) \mid \leq 1$$

a) Une itération Gauss-Seidel croissante :
Le schéma discrétisé s'écrit alors :

$$(D^n + E^n)\delta U^{n+1} = \widehat{\delta U} - F^n X^0 = \widehat{\delta U} \tag{16}$$

Avec comme notations suivantes : $M^n = E^n + D^n + F^n$ où
$D^n = (1 + \sigma|c|)_{1\leq j\leq N}$ est la matrice diagonale,
$E^n = (-\sigma c^+)_{1\leq j\leq N}$ est la matrice triangulaire inférieure stricte,
$F^n = (\sigma c^-)_{1\leq j\leq N}$ est la matrice triangulaire supérieure stricte.
et X^0 est le vecteur initiallisant la méthode; on a posé $X^0 = 0$.

Nous obtenons par Fourier après simplifications :

$$(-\sigma c^+ \exp(-i\xi_k) + 1 + \sigma \mid c \mid)(g_k(\Delta t) - 1) = \sigma c^+ \exp(-i\xi_k) - \sigma \mid c \mid -\sigma c^- \exp(i\xi_k)$$

Comme $1 + \sigma \mid c \mid -\sigma c^+ \cos\xi_k > 0$, On obtient :

$$g_k(\Delta t) = \frac{1 - \sigma c^- \exp(i\xi_k)}{1 + \sigma \mid c \mid -\sigma c^+ \exp(-i\xi_k)}$$

Nous posons $\mu = \mid c \mid \dfrac{\Delta t}{\Delta x}$: nombre de Courant-Friedrichs-Lewy et nous considérons les deux cas suivants :

Si $c > 0$: On vérifie aisément qu'on a à Δx fixé :

$$\lim_{\Delta t\longrightarrow\infty} \mid g_k(\Delta t) \mid = \sqrt{\frac{1}{1 + \mu(\mu+1)(1 - \cos\xi_k)}} = 0 \qquad (a)$$

et $\max\limits_{k\in[1;N]} \mid g_k(\Delta t) \mid \leq 1 \quad (b)$.

Si $c < 0$:

$$\lim_{\Delta t\longrightarrow\infty} \mid g_k(\Delta t) \mid = \sqrt{\frac{1 + \mu(\mu - 2\cos\xi_k)}{(1+\mu)^2}} = 1 \qquad (a')$$

et $\max_{k\in[1;N]} | g_k(\Delta t) | \leq 1$ $\qquad (b')$.

On déduit de (a) et (a') que le schéma est inconditionnellement stable pour tout c.

Nous supposons que Δt est grand. Pour c positif, la propriété (b) implique que la méthode converge en une relaxation. Pour c négatif, on est dans un cas limite : la méthode ne converge plus car le facteur d'amplification est proche de 1 (b').

b) Une itération Gauss-Seidel décroissante
Avec les mêmes notations qu'en a) nous obtenons :

$$(D^n + F^n)\delta U^{n+1} = \widehat{\delta U} - E^n X^0 = \widehat{\delta U} \tag{17}$$

D'une manière analogue au cas précédent, le schéma est inconditionnellement stable pour tout nombre c. Lorsque le pas de temps est infini, il ne converge pas pour $c > 0$ et converge en une relaxation pour $c < 0$.

Pour prendre en compte le signe de c, nous étudions l'algorithme sur deux balayages.

c) Deux itérations Gauss-Seidel : croissante-décroissante

Le schéma s'écrit :

$$\begin{cases} (D^n + E^n)X^1 = \widehat{\delta U} - F^n X^0 = \widehat{\delta U} \ : & \text{itération croissante (*)} \\ (D^n + F^n)\delta U^{n+1} = \widehat{\delta U} - E^n X^1 = D^n X^1 \ : & \text{itération décroissante (**)} \end{cases} \tag{18}$$

Considérons les deux cas suivant :

$c > 0$: De l'expression de (16) on a $F^n \equiv 0$, ce qui entraîne pour l'itération (**) :

$$\delta U^{n+1} = X^1$$

Soit en reportant cette condition dans (*), nous obtenons le schéma *croissant* (16) qui est stable et convergent.

$c < 0$: Cette fois $E^n \equiv 0$, ce qui implique pour (*) $D^n X^1 = \widehat{\delta U}$. Ainsi (**) s'écrit :

$$(D^n + F^n)\delta U^{n+1} = \widehat{\delta U}$$

Cette équation est indépendante de X^1 : c'est l'équation du schéma *décroissant* (17) qui possède alors les *bonnes* propriétés pour $c < 0$.

En somme, une seule itération réalisée dans le *bon sens* suffit à assurer la stabilité pour les grands pas de temps. Pour les systèmes, les valeurs propres peuvent être de signes différents donc on ne peut déterminer le sens de la relaxation ; seul le schéma (18) est stable.

Dans le cas non linéaire (Equations d'Euler par exemple), on pourra utiliser à chaque pas de temps le schéma (18) pour résoudre le système linéaire (14). On montrera à travers les expériences numériques, qu'une convergence linéaire très partielle (une itération (18) par exemple) suffit pour assurer la convergence vers la solution stationnaire du problème linéaire. Cependant la convergence non linéaire sera d'autant plus accélérée qu'on aura poussé la résolution des systèmes linéaires par un grand nombre d'itérations de type (18).

La méthode de Jacobi :

Bien que la méthode de Jacobi soit moins efficace que celle de Gauss-Seidel (voir Varga [26]), nous l'utiliserons cependant ici car elle présente deux avantages :

- Elle se combine bien à une méthode de stockage partiel de la matrice du système linéaire (voir Partie III.).

- Elle s'adaptate aisément au calcul vectoriel comme nous le montrerons dans la partie IV.

Nous faisons l'analyse de Fourier sur une seule itération : la discrétisation de l'équation de propagation nous conduit au système suivant avec les mêmes notations que précédemment :

$$D^n \delta U^{n+1} = \widehat{\delta U} - (E^n + F^n) X^0 = \widehat{\delta U} \tag{19}$$

Soit par Fourier après simplifications :

$$(1 + \sigma \mid c \mid)(g_k(\Delta t) - 1) = \sigma c^+ \exp(-i\xi_k) - \sigma \mid c \mid - \sigma c^- \exp(i\xi_k)$$

D'où comme $1 + \sigma \mid c \mid > 0$, nous obtenons explicitement le facteur d'amplification :

$$g_k(\Delta t) = \frac{1 + \sigma \mid c \mid \cos \xi_k - i\sigma c \sin \xi_k}{1 + \sigma \mid c \mid}$$

Et :

$$\mid g_k(\Delta t) \mid = \sqrt{\frac{1 + \mu(\mu + 2\cos \xi_k)}{(1 + \mu)^2}} \longrightarrow 1 \ ; \ \mu \to +\infty$$

et $\max_{k\in[1;N]} \mid g_k(\Delta t) \mid \leq 1.$

Donc la méthode de Jacobi sur une itération est aussi inconditionnellement stable mais elle ne converge pas indépendamment du signe de c. En particulier, il faudra beaucoup plus de balayages par pas de temps pour obtenir la convergence vers la solution stationnaire. De plus, lorsqu'une seule itération de Jacobi est appliquée, il n'est pas consistant : c'est ce que montre l'équation équivalente du schéma (19) :

$$U_t + (1-\mu)cU_x = \frac{|c|\Delta x}{2}(1-\mu)U_{xx}$$

Nous remarquons aussi que le fait d'initialiser la méthode de Jacobi par le vecteur nul permet de ne pas utiliser les termes extra-diagonaux ; le schéma (19) devient une méthode de préconditionnement diagonal que nous détaillerons plus loin dans la Partie III.

2.4. *Approximation spatiale en 2-D*

Volumes Finis :

La méthode utilisée ici est introduite en [5] et [4] : elle est de type *Volumes-Finis* avec des cellules construites à partir d'éléments finis triangulaires. Par ailleurs nous utilisons dans ce paragraphe les notations précédentes.

Soit une triangulation $\mathcal{T}_h$ sur un polygône Ω_h domaine de calcul approchant le domaine Ω vérifiant :

$$\overline{\Omega}_h = \bigcup_{j=1}^{nt} T_j \ ; \ T_j \in \mathcal{T}_h$$

Où les T_j sont les éléments de la triangulation i.e.triangles, nt est le nombre total de triangles de $\mathcal{T}$ et h dénote la longueur maximale des côtés des triangles.

A chaque sommet a_i $(i = 1, ..ns)$ est associée une cellule C_i (voir Figure **II.**) avec ns nombre total des nœuds. La cellule C_i se construit en joignant successivement les centres de gravité des triangles contenant le nœud a_i et les milieux entre le nœud a_i et son voisin :

Les cellules forment alors un autre recouvrement de $\overline{\Omega}_h$

$$\overline{\Omega}_h = \bigcup_{i=1}^{ns} C_i$$

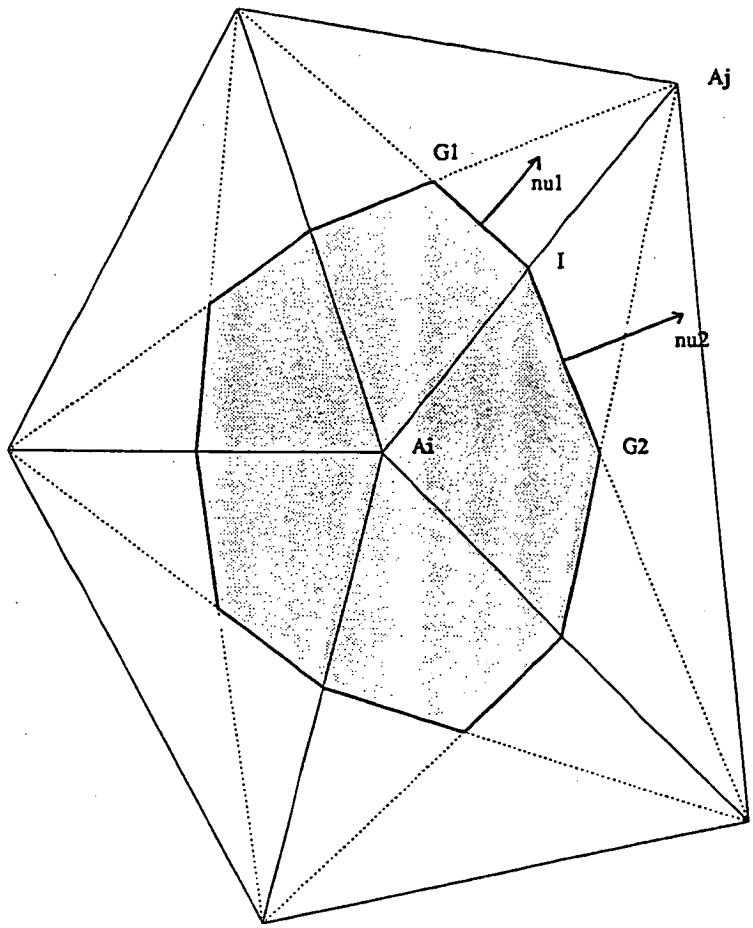


Fig. II : *Cellule d'intégration* C_i

Nous recherchons les solutions faibles du problème continu (1) des équations d'Euler. Nous approchons le problème continu défini sur $\Omega \times I\!R^+$ par le problème discret défini sur $\Omega_h \times [0,T]$. L'intégration en temps est réalisée par la méthode implicite décrite au §2. Soit $W_i^n = W(a_i(x,y),t^n)$. Alors on obtient :

$$\iint_{\Omega_h} (\frac{W^{n+1} - W^n}{\Delta t^n} + F(W^{n+1})_x + G(W^{n+1})_y)\varphi \ = \ 0 dx dy$$

et $W^0 = W_\infty$.

Prenons les fonctions caractéristiques des cellules pour les fonctions tests φ :

$$mes(C_i)\frac{W_i^{n+1} - W_i^{n+1}}{\Delta t^n} + \sum_{j \in K(i)} \int_{\partial C_i \cap \partial C_j} \Phi_{ij}^{n+1} d\sigma + \int_{\partial \Omega_h \cap \partial C_i} \Psi_i^{n+1} d\sigma = 0 \quad (20)$$

Nous avons noté par $mes(C_i)$ l'aire de la cellule C_i, ∂C_i est la fontière de la cellule et $K(i)$ est l'ensemble des nœuds voisins à a_i. Φ_{ij} et Ψ_i sont les flux numériques utilisés et sécrits plus loin.

Notations des flux et dérivées de flux :

Nous posons $\vec{\eta}_{ij} = \int_{\partial C_i \cap \partial C_j} \vec{\nu} d\sigma$ avec $\vec{\nu}$ vecteur normal sortant à C_i.

Φ_{ij}^{n+1} est le flux numérique implicite linéarisé et simplifié défini au §2.1 échangé entre les deux cellules C_i et C_j :

$$\Phi_{ij}^{n+1} = \mathcal{A}^+(W_i^n, \vec{\eta}_{ij}) W_i^{n+1} + \mathcal{A}^-(W_j^n, \vec{\eta}_{ij}) W_j^{n+1}$$

La décomposition est celle du flux de Steger-Warming
$\mathcal{A}^\pm(W, \vec{\eta}) = (A(W)\eta^x + B(W)\eta^y)^\pm$ (voir §1) où $\vec{\eta} = \begin{pmatrix} \eta^x \\ \eta^y \end{pmatrix}$.

Nous notons le flux Euler par $\mathcal{F}(W, \vec{\eta}) = F(W)\eta^x + G(W)\eta^y$.

Calcul sur $\partial\Omega$:

Enfin, le calcul sur les bords du domaine $\partial\Omega$ se décompose en deux parties :

sur Γ_∞ (bord infini), nous calculons un flux décentré de Steger-Warming entre la valeur W_i^n au nœud a_i de la cellule C_i et sa valeur imposée W_∞. Ce flux se linéarise uniquement sur la valeur W_i.

Sur Γ_B (bord glissant), le flux se construit à partir de la condition de glissement : $\vec{V} \cdot \vec{\nu} \ = \ 0$ où $\vec{V}$ est la vitesse de fluide et $\vec{\nu}$ est le vecteur normal sortant du domaine Ω. Notons par $\mathcal{F}_B$ ce flux :

$$\mathcal{F}_B(W,\vec{\nu}) = \begin{pmatrix} 0 \\ p\nu^x \\ p\nu^y \\ 0 \end{pmatrix}$$

Ce flux se différencie et on obtient son jacobien $\mathcal{A}_B$:

$$\mathcal{A}_B(W,\vec{\nu}) = (\gamma-1)\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2}(u^2+v^2)\nu^x & -u\nu^x & -v\nu^x & \nu^x \\ \frac{1}{2}(u^2+v^2)\nu^y & -u\nu^y & -v\nu^y & \nu^y \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

<u>δ-schéma :</u>

De façon analogue au cas 1-D (§2.1.); on en déduit après linéarisation et simplification de l'équation (20), le schéma d'ordre un à deux phases en formulation δ-schéma :

1 : Phase physique

$$\widehat{\delta W_i} = -\frac{\Delta t^n}{mes(C_i)}\left\{\sum_{j\in K(i)}\frac{1}{2}\left(\mathcal{F}(W_i^n,\vec{\eta}_{ij})+\mathcal{F}(W_j^n,\vec{\eta}_{ij})-\int_{W_i^n}^{W_j^n}|\mathcal{A}(W,\vec{\eta}_{ij})|\,dW\right)\right.$$
$$\left.+\int_{\Gamma_\infty\cap\partial C_i}\mathcal{A}^+(W_i^n,\vec{\nu}_i)W_i^n+\mathcal{A}^-(W_\infty,\vec{\nu}_i)W_\infty d\sigma+\int_{\Gamma_B\cap\partial C_i}\mathcal{F}_B(W_i^n,\vec{\nu}_i)d\sigma\right\} \qquad (21)$$

2 : Phase mathématique

$$\frac{mes(C_i)}{\Delta t}\delta W_i^{n+1} + \sum_{j\in K(i)} (\mathcal{A}^+(W_i^n, \vec{\eta}_{ij})\delta W_i^{n+1} + \mathcal{A}^-(W_j^n, \vec{\eta}_{ij})\delta W_j^{n+1})$$
$$+ \int_{\partial C_i \cap \Gamma_\infty} \mathcal{A}^+(W_i^n, \vec{\nu}_i)\delta W_i^{n+1} + \int_{\partial C_i \cap \Gamma_B} \mathcal{A}_B(W_i^n, \vec{\nu}_i)\delta W_i^{n+1} \qquad (22)$$
$$= \frac{mes(C_i)}{\Delta t}\widehat{\delta W_i}$$

$$W_i^{n+1} = W_i^n + \delta W_i^{n+1}$$

Stockage des termes implicites :

Le calcul de W^{n+1} à partir de W^n nécessite la résolution d'un système linéaire à chaque pas de temps. Nous l'écrivons de manière simplifiée :

$$D_i^n \delta W_i^{n+1} + \sum_{j\in K(i)} E_{ij}^n \delta W_j^{n+1} = \widehat{\delta W_i^n}$$

Nous avons noté par D_i^n les termes matriciels des blocs 4×4 diagonaux,et par E_{ij}^n les termes matriciels des blocs extra-diagonaux, pour chaque cellule C_i.

Dans tous les problèmes de type éléments finis, la stratégie du stockage des termes implicites est prépondérante. Le stockage complet de la matrice est de $16 \times NS^2$ (NS nombre total de points); dès que NS est plus grand que 2000 la mémoire des plus grands ordinateurs actuels (CRAY-2 par exemple) est dépassée. Même avec un stockage de profil et une renumérotation des nœuds adéquate, le stockage reste prohibitif.

La matrice du système linéaire est très *creuse* : en effet le cardinal de $K(i)$ est petit devant le nombre total de nœuds, c'est pourquoi nous effectuons un stockage *Morse* où seuls les termes non nuls sont stockées. La matrice n'étant pas symétrique le stockage est réalisé de la manière suivante :

– Nous considérons à part le stockage des blocs diagonaux D_i^n :
$4^2 \times card\{a_i\}_{i=1,NS} = 16 \times NS$ (le nombre 4 étant le nombre de variables indépendantes des équations d'Euler).

– Pour le stockage des blocs non diagonaux,l'assemblage des flux est réalisé sur les couples de nœuds voisins $[a_i, a_j]$ (assemblage dit **par segments**). Le calcul et le stockage par nœud est beaucoup plus coûteux puisque l'on recalcule deux fois les

mêmes flux (Un bord de cellule est commun à deux cellules). Sur chaque segment $[a_i, a_j]$ il y a la contribution de flux de a_i vers a_j : E^n_{ij}, et, inversement celle de a_j vers a_i : E^n_{ji}. On évalue le nombre de termes extra-diagonaux :
$2 \times 4^2 \times card\{[a_i, a_j]\}_{i=1,NS\ ;\ j\ voisin\ de\ i} = 32 \times NSEG$ avec $NSEG$ nombre total des segments $[a_i, a_j]$ estimé à $3 \times NS$ dans le cas d'un maillage assez régulier.

Nous obtenons au total un encombrement mémoire matriciel d'environ $112 \times NS$. Nous avons estimé à $100 \times NS$ le stockage des autres variables pour un schéma explicite d'ordre 1. Dans ce cas, on peut atteindre un stockage limite pour un maillage d'environ 400000 points sur CRAY-2 et de 50000 points pour un IBM 3081. Par conséquent, le stockage de la matrice est un obstacle à la réalisation d'un calcul à grand nombre de nœuds : pour des géométries 3-D surtout où le stockage est encore plus limitatif. (Voir la Partie III. pour le traitement de ce problème de mémoire).

3. Schémas d'ordre supérieur en espace

Le schéma implicite décrit en (21)-(22) est précis à l'ordre un en temps et en espace. Pour améliorer la précision spatiale des solutions discrètes stationaires des équations (1), il suffit d'augmenter l'ordre de précision de l'approximation dans la phase explicite du schéma; la phase implicite restant à l'ordre un.

Mais on sait que les méthodes d'ordre deux ne vérifient pas le principe de monotonie, les solutions présentent des oscillations surtout près des chocs et discontinuités. Nous utiliserons des approximations de type M.U.S.C.L. introduites par van Leer (*Monotone Upwind Scheme for Conservative Laws*) [11].

Cette méthode repose sur deux idées de base :

1) On augmente la precision en élévant le degré de l'interpolation de l'inconnue W. On part d'une interpolation constante par cellule P^0 (conduit à un schéma d'ordre un) à une interpolation P^1 linéaire par cellule. On aura ainsi des valeurs plus précises de l'inconnue aux interfaces des cellules ; cependant la fonction de flux numérique reste identique, seuls ses arguments sont changés.

2) Des techniques de limiteurs sont alors introduites pour atténuer les oscillations numériquues qui peuvent de produire. Nous utiliserons des limiteurs de pente que nous détaillerons dans 3.2.

3.1. *Etude monodimensionnelle d'une classe β de flux*

Des investigations sur les schémas linéarisés d'ordre deux sont décrites dans [18] avec comme modèle les équations d'Euler 1-D en comparant diverses linéarisations

de flux : par décomposition *plus-moins* et par différenciation. Une analyse plus poussée, par le calcul algébrique, a été réalisée par J.A. Désideri [9] et montre certains effets néfastes pouvant gêner la convergence des schémas β implicites d'ordre deux.

Nous utilisons l'analyse de Fourier avec les notations du §2.1.5. Elle nous a permis d'obtenir des résultats théoriques assez précis sur l'équation de propagation scalaire linéaire.

3.1.1. *Flux d'ordre deux 1-D*

Avec les mêmes notations du §2.1. nous décomposons le flux explicite de la manière suivante :

$$\widehat{\delta W_i} = -\sigma(\Phi^n_{i+\frac{1}{2}} - \Phi^n_{i-\frac{1}{2}})$$

$$\Phi^n_{i+\frac{1}{2}} = \Phi(W^n_{i+\frac{1}{2}-}, W^n_{i+\frac{1}{2}+})$$
$$\Phi^n_{i-\frac{1}{2}} = \Phi(W^n_{i-\frac{1}{2}-}, W^n_{i-\frac{1}{2}+})$$
$$\sigma = \frac{\Delta t}{\Delta x}$$

Les différentes valeurs $W_{i\pm\frac{1}{2}\pm}$ sont évaluées par une interpolation de type P_1 (fonctions linéaires sur les intervalles : $]x_i - \frac{\Delta x}{2}; x_i + \frac{\Delta x}{2}[$). Nous obtenons alors :

$$W_{i+\frac{1}{2}-} = W_i + \frac{\Delta x}{2} P_{i+\frac{1}{2}-}$$
$$W_{i+\frac{1}{2}+} = W_{i+1} - \frac{\Delta x}{2} P_{i+\frac{1}{2}+}$$

Nous introduisons le paramètre réel β pour calculer les *pentes* $P_{i+\frac{1}{2}-}$ et $P_{i+\frac{1}{2}+}$ avec $0 \leq \beta \leq 1$:

$$P_{i+\frac{1}{2}-} = (1-\beta)(\frac{W_{i+1} - W_i}{\Delta x}) + \beta(\frac{W_i - W_{i-1}}{\Delta x})$$
$$P_{i+\frac{1}{2}+} = (1-\beta)(\frac{W_{i+1} - W_i}{\Delta x}) + \beta(\frac{W_{i+2} - W_{i+1}}{\Delta x})$$

Si $\beta = 0$ les pentes sont centrées en $x_{i+\frac{1}{2}}$. Pour $\beta = 1$ les pentes sont totalement décentrées.

Ce schéma numérique sera referé par la suite comme le β *-schéma.* Dans le cas où le jacobien A est constant, il est équivalent à l'ordre trois au schéma suivant :

$$AW_x = -\frac{(1-3\beta)}{6} AW_{xxx}\Delta x^2 - \frac{\beta}{4} \mid A \mid W_{xxxx}\Delta x^3$$

Le schéma est donc au moins précis à l'ordre deux en espace et à l'ordre un en temps. De plus pour $\beta = \frac{1}{3}$ il est précis à l'ordre trois en espace. Le cas $\beta = 0$ est le cas particulier du schéma d'ordre deux centré à trois points. Le cas $\beta = 1$ représente le schéma où le flux est le plus décentré.

3.1.2. *Stabilité linéaire*

Nous utilisons l'équation de propagation $U_t + cU_x = 0$ avec des conditions de périodicité en espace (pour les notations, voir le §2.4.).

La discrétisation du β-Schéma implicite conduit à la résolution d'un système linéaire $M^n \delta U^{n+1} = \widehat{\delta U}$.

La phase physique s'écrit :

$$\widehat{\delta U_i} = -\sigma \left\{ (1-\beta) c\delta^0 + \beta c^- \delta^+ + \beta c^+ \delta^- \right\} \tag{23}$$

avec :

$$\begin{cases} \delta^0 = \dfrac{U^n_{i+1} - U^n_{i-1}}{2} & \text{opérateur centré} \\ \delta^+ = \dfrac{-U^n_{i+2} + 4U^n_{i+1} - 3U^n_i}{2} & \text{opérateur décentré à droite} \\ \delta^- = \dfrac{3U^n_i - 4U^n_{i-1} + U^n_{i-2}}{2} & \text{opérateur décentré à gauche} \end{cases}$$

La matrice implicite d'ordre un en espace s'écrit :

$$M^n = Tridiag.(\ -\sigma c^+ \ ; \ 1 + \sigma \mid c \mid \ ; \ \sigma c^- \ )$$

On obtient par l'analyse de Fourier :

$$(1 + \sigma \mid c \mid (1 - \cos\xi_k) + i\sigma c \sin\xi_k) g_k(\Delta t) =$$
$$1 + \sigma \mid c \mid (1 - \cos\xi_k) - \sigma \mid c \mid \beta(1 - \cos\xi_k)^2 - i\sigma c\beta \sin\xi_k (1 - \cos\xi_k)$$

Comme $1 + \sigma \mid c \mid (1 - \cos\xi_k) > 0$ alors :

$$g_k(\Delta t) = \frac{1 + \sigma \mid c \mid (1 - \cos\xi_k) - \sigma \mid c \mid \beta(1 - \cos\xi_k)^2 - i\sigma c\beta \sin\xi_k (1 - \cos\xi_k)}{\cdot 1 + \sigma \mid c \mid (1 - \cos\xi_k) + i\sigma c \sin\xi_k} \tag{24}$$

Nous posons $\mu = \sigma \mid c \mid$ et calculons le module du facteur d'amplification :

$$\mid g_k(\Delta t) \mid = \sqrt{\frac{1 + 4\mu \sin^2 \frac{\xi_k}{2}(1 - 2\beta \sin^2 \frac{\xi_k}{2} + \mu \sin^2 \frac{\xi_k}{2}(1 - 4\beta(1-\beta)\sin^2 \frac{\xi_k}{2}))}{1 + 4\mu(\mu+1)\sin^2 \frac{\xi_k}{2}}}$$

Proposition 4 :

Le schéma (22) est inconditionnellement l_2-Stable.

Démonstration :

En effet, vérifions que pour tout mode $k \in [1, N]$ la condition de von Neumann:

$$\mid g_k(\Delta t) \mid \leq 1 \iff$$

$$1 + 4\mu \sin^2 \frac{\xi_k}{2}(1 - 2\beta \sin^2 \frac{\xi_k}{2} + \mu \sin^2 \frac{\xi_k}{2}(1 - 4\beta(1-\beta)\sin^2 \frac{\xi_k}{2})) \leq 1 + 4\mu(\mu+1)\sin^2 \frac{\xi_k}{2}$$

Soit après simplications :

$$2\beta(1 - \cos \xi_k) + \mu\left((1 + \cos \xi_k) + 2\beta(1-\beta)(1 - \cos \xi_k)^2\right) \geq 0$$

ce qui est toujours vrai pour tout $k \in [1, N]$, $0 \leq \beta \leq 1$ et $\forall \mu \geq 0$.

$\oslash$

3.1.3. *Etude sur les grands pas de temps*

Lorsque $\Delta t \longrightarrow +\infty$ (le pas en espace etant fixé) on a :

$$\mid g_k(\Delta t) \longrightarrow \sqrt{\frac{(1 - \cos \xi_k)}{2}(1 - 2\beta(1-\beta)(1 - cos\xi_k)} = f(\beta)$$

L'étude de la fonction $f(\beta)$ montre que son minimun est atteint en $\beta = \frac{1}{2}$:

$$f(\frac{1}{2}) = \frac{|\sin \xi_k|}{2} \leq \frac{1}{2}$$

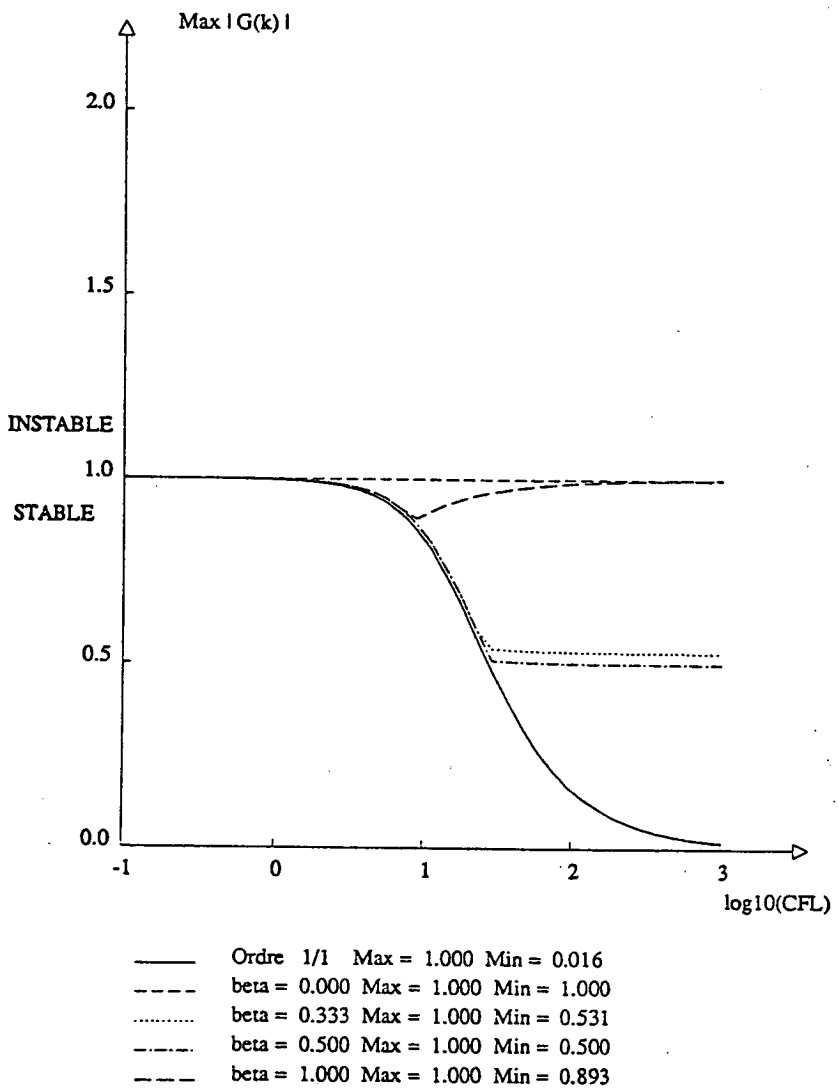


Fig. III : *Courbes de stabilité des schémas β*

Nous posons : $\alpha = \max_{k\in[1,N]} f(\beta)$. Considérons la fonction d'erreur $\epsilon(t)$:

$$\epsilon(n\Delta t) = \frac{\|U^{n+1} - U^n\|}{\|U^1 - U^0\|}$$

Lorsque Δt est grand, on a $\epsilon(n\Delta t) \simeq \alpha$. La valeur de α est plus grande que $0,5$, la

convergence est donc ralentie : la fonction d' erreur devient constante indépendante du pas de temps (lorsque n est grand), ceci, n'est pas le cas du schéma d'ordre 1 (courbe en trait plein) où $\alpha \simeq 0$ lorsque $\Delta t \rightarrow +\infty$. Une valeur optimale est obtenue pour les schémas d'ordre 2 avec $\beta = 1/2$: $\alpha = 0,5$.

Comme nous le montre la Figure **III.**, les schémas implicites β se divisent en deux classes distinctes :
– Schémas à convergence lente : c'est le cas de $\beta = 1$ et $\beta = 0$ où $\alpha = 1$. Pour ces schémas, l'utilisation des grands pas de temps n'est pas optimale. Ces deux cas particuliers ont été analysés plus en détail par J.A. Désidéri [9].
– Schéma à convergence rapide pour $\beta \simeq \frac{1}{2}$. Les grands pas de temps favorisent la convergence sans qu'elle soit toutefois quadratique.

Pour le cas particulier du schéma d'ordre 3 ($\beta = \frac{1}{3}$), on montre aisément que $\alpha = \frac{3\sqrt{2}}{8} \simeq 0,5303301....$ Cette valeur est très proche de la valeur optimale ($\alpha = 0,5$).

3.2. *Calcul des pentes en 2-D*

Dans le cas des équations d'Euler bidimensionnelles, le gain en précision du flux explicite produit des oscillations près des discontinuités, surtout pour des écoulements à grand nombre de Mach (supersoniques). Pour y remédier nous introduisons des limiteurs de pente comme en [4], [11] qui régularisent la solution.

Comme dans le cas mondimensionnel, pour atteindre le second ordre nous utilisons une interpolation linéaire de type P_1-Galerkin [1] des fonctions à approcher. Sur chaque segment $[a_i, a_j]$ nous calculons les gradients :

$$\overrightarrow{\nabla W_i} = \frac{1}{mes(C_i)} \iint_{C_i} \overrightarrow{\nabla W} dxdy$$

D'où :

$$\begin{cases} W_{ij} = W_i + \dfrac{\overrightarrow{\nabla W_i}}{2} \cdot \overrightarrow{ij} \\ W_{ji} = W_j - \dfrac{\overrightarrow{\nabla W_j}}{2} \cdot \overrightarrow{ij} \end{cases}$$

La procédure de limitation est réalisée par triangles [4]. Nous posons :

$$\overrightarrow{\nabla W_T} = \begin{pmatrix} W_{T,x} \\ W_{T,y} \end{pmatrix}$$

La notation $W_{T,x}$ signifie que la dérivée de W est restreinte au triangle T avec :

$$\begin{cases} W_{T,x} = \sum_{k=1}^{3} W_k \ \varphi_{T,x}(k) \\ W_{T,y} = \sum_{k=1}^{3} W_k \ \varphi_{T,y}(k) \end{cases}$$

Les fonctions $\varphi_{T,x}(k)$; $\varphi_{T,y}(k)$ représentent les dérivées de fonctions de base des éléments finis P_1 en a_k.

On cherche un triangle T_0 tel que :

$$W_{T_0,x} = \min_{j,\ T_j \cap C_i \neq \emptyset} |W_{T_j,x}|$$

On pose : $W_x^{lim} = W_{T_0,x}$ si W_x est monotone, sinon $W_x^{lim} = 0$.

Le calcul de W_y^{lim} est identique à W_x^{lim} (le procédé peut aussi se généraliser en 3-D). Enfin nous posons :

$$\overrightarrow{\nabla W_i^{lim}} = \begin{pmatrix} W_x^{lim} \\ W_y^{lim} \end{pmatrix}$$

Et nous obtenons les valeurs interpolées :

$$\begin{cases} W_{ij} = W_i + \dfrac{\overrightarrow{\nabla W_i^{lim}}}{2} \cdot \overrightarrow{ij} \\ W_{ji} = W_j - \dfrac{\overrightarrow{\nabla W_j^{lim}}}{2} \cdot \overrightarrow{ij} \end{cases}$$

Remarque :

Le calcul des pentes présenté ici correspond au cas $\beta = 0,5$ du paragraphe précédent. Il est possible de calculer d'autres pentes plus précise comme en [10]. Les pentes sont alors une combinaison avec des gradients calculés au milieu du segment $[a_i, a_j]$ et des gradients calculés dans des triangles alignés avec le segment.

L'introduction des limiteurs peuvent affecter la précision d'ordre 2 des schémas, mais ceux-ci n'interviennent que lorsque apparaissent des oscillations c'est à dire près des chocs et discontinuités. Pour déterminer avec plus de précision les zones de chocs il est nécessaire d'augmenter le nombre de points dans ces zones par raffinements locaux.

4. Résultats numériques

Nous présentons des calculs pour des écoulements bidimensionnels.

4.1. *Ordre 1*

L'étude est réalisée sur des écoulements transsoniques pour lesquels nous avons pu ajuster les nombres de Courant de façon à obtenir des convergences rapides.

a) Convergence quasi-quadratique à $CFL = 1000$:

Nous présentons le test du canal avec un dos d'âne d'une hauteur de 4,2%. La maillage discret compte 1512 points. Le calcul est réalisée en régime transsonique : $M_\infty = 0,85$ (Figure 1). Les paramètres intervenant pour ce calcul sont le nombre de Courant imposé à 1000 et le nombre d'itérations Gauss-Seidel appliqué lors de la résolution de la phase mathématique, qui est de 50, ce qui fournit à une bonne approximation de la solution du système linéaire. On observe une convergence est dès les premiers pas de temps, on retrouve alors des résultats comparables à ceux obtenus par de B. Stoufflet [2] avec un schéma voisin : on obtient une erreur résiduelle normalisée de 10^{-4} en 7 itérations : ce comportement favorable, comparable à celui d'une méthode de Newton, peut s'expliquer en constatant que la condition initiale par écoulement uniforme est assez proche de la solution stationnaire.

Par ailleurs, nous présentons des résultats obtenus avec très peu d'itérations linéaires : 2 itérations Gauss-Seidel ou 4 itérations Jacobi, mais la convergence non linéaire est sensiblement ralentie.

La Courbe 2.1 avec la méthode de Gauss-Seidel et la Courbe 2.2 avec celle de Jacobi, comparent les convergences non linéaires en fonction du nombre d'itérations linéaires.

Du point de vue du coût, c'est à dire le temps utilisé pour un ordinateur pour atteindre *la quasi-convergence* de la solution stationnaire en divisant l'erreur résiduelle par 1000, nous remarquons l'existence d'un nombre optimal d'itérations linéaires à effectuer à chaque pas de temps qui se situe vers 15 avec Gauss-Seidel :

L' efficacité des méthodes itératives est représentée sur la Courbe 2.3.

b) Convergence en deux phases :

Dans le cas où la condition initiale est relativement loin de la solution stationnaire (lorsque la géométrie est plus complexe ou en présence d'un point d'arrêt par exemple), nous ne pouvons pas faire des calculs avec des grands CFL dès

les premières itérations, c'est la raison pour laquelle on introduit deux phases dans l'évolution des calculs : Nous avons une première phase avec des petits CFL qui correspond à la recherche de la solution sur quelques itérations (de 3 à 10 itérations). Les nombres de CFL apliqués alors sont données par une suite croissante linéaire : $CFL_n = n^\alpha$(n est le numéro de l'itération et α une constante à régler). Puis une seconde phase intervient avec des grands CFL ; la convergence quadratique devient alors possible . La progression du CFL vaut : $CFL_n = \frac{K}{RES^n}$ avec RES^n est la fonction d'erreur résiduelle à l'instant n. (voir §2.3.) K est une constante ajustable (de l'ordre de l'unité). Cette approche est voisine de celle de de B. van Leer et W A. Mulder [23] sur l'adaptation des pas de temps liée aux méthodes de relaxation pour un problème hyperbolique. Les tests sont réalisés sur le profil d'une aile d'avion en régime transsonique : *NACA 0012* avec 800 points (figure 3). Ce calcul diverge lorsque le CFL vaut 1000 dès les premières itérations : en introduisant la montée progressive du CFL. La fonction d'erreur a été divisée par 1000 en 8 itérations.

La figure 4.1 montre la convergence en ordre 1 sur le Naca0012.

4.2. *Ordre 2*

a) Comparaisons avec l'ordre 1:

Les suites de nombre de CFL reste en tous points identiques avec l'ordre 1. Nous remarquons que la convergence reste suffisamment rapide , l'ordre 2 ne l'affecte pas sensiblement. Sur les différentes géométries précédentes nous obtenons une perte d'efficacité de l'ordre de 1,5 : convergence en 10 itérations (en divisant par 1000 le résidu) , ce qui est peu compte tenu du gain appréciable de l'ordre de précision en espace qui est plus élevé. Donc nous pouvons dire que la matrice implicite décentrée d'ordre 1, préconditionne bien le flux explicite d'ordre 2.

La comparaison ordre 1 / ordre 2 pour un écoulement autour d'un profil NACA 0012 avec un CFL fonction du résidu est représentée sur la courbe 4.2

b) Ecoulement supersonique à grand nombre de Mach

Nous étudions le problème du corps arrondi émoussé à l'avant, discrétisé avec un maillage avec 1908 points à $M_\infty = 8$ (Figure 5.1), nous remarquons que nous pouvons pas utiliser des grands pas de temps pendant les 50 premières itérations sans faire apparaître d'instabilité. En effet, la solution initiale très régulière puisque uniforme est loin de la solution stationnaire très discontinue (présence d'un choc). En ordre deux, les limiteurs préviennent de l'apparition de fortes oscillations au niveau des discontinuités.

Les Courbes 6 montrent l'évolution la convergence pour le corps émoussé (1908 pts). La courbe 6.1 est un calcul à l'ordre 1, la courbe 6.2 est l'ordre 2 : la convergence n'est pas sensiblement ralentie. La courbe 6.2 montre le ralentissement de convergence lorsqu'on place le corps émoussé à 30 degré d'incidence : le calcul est d'ordre 2 avec les limiteurs de pentes, le CFL atteint est 30. L'erreur résiduelle relative a été divisée par 1000 en 100 itérations dans le cas non incident et en 170 itérations dans le cas fortement incident. Pour les solutions obtenues (isovaleurs du Mach et déviation de l'entropie sur le corps) voir les figures 5.2 et 5.3 (ordre 1 avec incidence), 5.4 et 5.5 (ordre 2 sans incidence), 5.6 et 5.7 (ordre 2 avec incidence).

5. Conclusion

Les méthodes implicites linéarisées peuvent être mises en œuvre pour tous les problèmes concernant les équations d'Euler ; notamment lors de problèmes plus difficiles ou complexes avec des points d'arrêts, en régime supersonique à nombre de Mach comme le corps émoussé en forte incidence ou bien le problème de l'écoulement autour d'un cylindre complet (voir [19]).

Mais plus le problème se complique, plus il nécessite un grand nombre de points de discrétisation. Ceci a pour conséquence d'augmenter sérieusement l'encombrement mémoire du code implicite, surtout en ce qui concerne la matrice du système linéaire. C'est pourquoi dans la partie suivante nous proposons des approches nécessitant moins de l'espace mémoire.

III. Schémas à faible stockage matriciel

1. Problème de l'encombrement mémoire

Le stockage de la matrice implicite est coûteux devant celui des autres tableaux stockées. En effet, comme il est expliqué dans la partie précédente, pour un code 2-D dans le cas d'un maillage assez régulier, nous estimons l'encombrement de la matrice (le stockage effectué est *Morse*) à $112 \times NS$ avec NS nombre total de sommets du domaine Ω_h devant $100 \times NS$ pour les autres variables.

Nous proposons deux algorithmes qui conservent uniquement les termes des blocs diagonaux, qui ont pour conséquence de diviser par 7 l'encombrement mémoire du stockage de la matrice. Nous rappelons que le stockage des termes non diagonaux étant réalisé par segments occupe $96 \times NS$.

Remarque : Dans le cas des maillages 3-D où les éléments sont des tetraèdres, le stockage complet de la matrice implicite est estimé à $350 \times NS$ (soit 3 fois plus important qu'en 2-D), le stockage matriciel pour un algorithme de stockage partiel de la matrice sera 14 fois moins important.

Pour un schéma d'ordre deux, les gradients des solutions sont calculés et utilisés uniquement lors de la phase physique. Par contre les blocs diagonaux n'interviennent que lors de la phase mathématique ; ceci permet le stockage des gradients et des blocs diagonaux sur une même variable (les longueurs des deux tableaux sont presque identiques). Ainsi l'encombrement mémoire d'un code implicite linéarisé, en ordre deux, est ramené à celui d'un code explicite.

2. Méthode de Jacobi sans stockage

Nous utilisons à nouveau les notations de la partie précédente : voir les phases physiques et mathématiques de l'algorithme avec stockage matriciel. Il s'agit de conserver les propriétés du schéma implicite en ce qui concerne la convergence mais en évitant le stockage des termes non diagonaux . Nous énonçons alors l'algorithme du calcul de la solution sur chaque itération non linéaire :

1 Phase physique : le calcul du flux explicite $\widehat{\delta W}$ reste inchangé. Il peut-être effectué en ordre un ou en ordre deux.

2 Phase mathématique :

a) Nous réalisons tout d'abord l'assemblage des blocs diagonaux D^n que nous inversons et stockons. Nous avons $M^n = D^n + E^n$ où M^n est la matrice implicite et E^n est la matrice des éléments non diagonaux.

b) Puis nous résolvons le système linéaire $M^n \delta W^{n+1} = \widehat{\delta W}$ par la méthode de Jacobi :

– La méthode est initialisée par X^0 :

$$X^0 = (D^n)^{-1} \widehat{\delta W}$$

– Puis nous calculons via une boucle sur les *segments* le second membre S de Jacobi à chaque itération :

$$S = \widehat{\delta W} - E^n X^{iter}$$

Où les E^n sont les blocs non diagonaux non stockés :
En effet le calcul de S se décompose comme suit :
(*) Initialisation : $S = \widehat{\delta W}$
(**) Pour chaque segment $[a_i, a_j]$ on assemble S_i et S_j :

$$\begin{cases} S_i = S_i - E^n_{ij} X^{iter}_j \\ S_j = S_j - E^n_{ji} X^{iter}_i \end{cases}$$

– On peut alors calculer (par sommets) l'itéré $iter + 1$ de Jacobi :

$$X^{iter+1} = (D^n)^{-1} S$$

– Pour $iter = itermax$ nous obtenons la solution partiellement convergée du système linéaire :

$$\delta W^{n+1} = X^{itermax}$$

c) calcul de la solution à l'instant $n + 1$:

$$W^{n+1} = W^n + \delta W^{n+1}$$

Il est clair que la phase mathématique sera plus coûteuse que celle du schéma implicite avec stockage complet, à cause du recalcul à chaque itération de Jacobi des termes non diagonaux : chaque itération coûte l'équivalent d'un calcul de flux complet. D'autre part la convergence de la méthode linéaire sera ralentie, la méthode de Jacobi étant moins rapide que celle de Gauss-Seidel à nombre égal d'itérations (voir le §2.4. de la Partie II). Ceci a pour conséquence d'augmenter le nombre d'itérations non linéaires pour obtenir la même convergence (le facteur estimé est de l'ordre de 2).

Cas particuliers et remarques :

Pour obtenir une solution quasiment convergée du système linéaire il faut augmenter considérablement le nombre de relaxations. Nous testons l'état de convergence du système linéaire en introduisant une fonction d'erreur $\epsilon(iter)$ normalisée :

$$\epsilon(iter) = \frac{\|X^{iter} - X^{iter-1}\|}{\epsilon(iter = 1)}$$

Nous estimerons avoir une solution quasi-convergée lorsque la fonction d'erreur aura atteint une valeur de l'ordre de 10^{-6}.

Dans le cas où $itermax = 0$ c'est à dire lorsque aucune itération de Jacobi n'est réalisée, le schéma global n'utilise plus les termes non diagonaux et donc il perd certaines propriétés du schéma implicite. Nous étudions ce cas particulier dans le prochain paragraphe.

3. Méthode à préconditionnement diagonal

3.1. *Description de la méthode*

Nous pouvons imaginer un algorithme qui n'utilise plus les termes extra-diagonaux de la matrice du système linéaire : cette méthode perd alors des certaines propriétés des schémas implicites précédents mais elle peut s'avérer plus efficace que les méthodes explicites classiques. Elle est notamment utilisée par A.Eberlé en [12], [13] pour des calculs très lourds (520 000 cellules) sur calculateur vectoriel. L'algorithme est plus simple que le précédent et s'écrit :

1 Phase physique : calcul du flux explicite $\widehat{\delta W}$

2 Phase mathématique :
a) Nous assemblons les blocs diagonaux dans D^n qui sont éventuellement corrigés : en effet, la matrice peut devenir singulière lors de changement de signes des valeurs propres à travers une cellule (voir [12]). Pour chaque bloc diagonal on calcule :

$$D_{corr} = D^n + I \,.\, |\lambda|_{max} \,(1 + sign(\epsilon|\lambda|_{max} - minD))$$

Avec λ première ou seconde valeur propre, $\epsilon = 0,01$ et $minD$ est le plus petit élément diagonal du bloc 4×4.

Enfin nous stockons les inverses de ces blocs.

b) Résolution par préconditionnement diagonal :

$$\delta W^{n+1} = (D^n)^{-1}\widehat{\delta W}$$

c) Calcul de la solution à l'instant $n+1$:

$$W^{n+1} = W^n + \omega \delta W^{n+1}$$

où ω est un paramètre de sous-relaxation destiné à augmenter la stabilité et la convergence (voir plus loin) dans la cas d'un flux explicite d'ordre égal à deux. Dans le cas de l'ordre 1 on prend $\omega = 1$.

3.2. *Etude de la stabilité linéaire*

Comme dans la partie précédente, nous examinons l'équation de propagation qui est périodique en espace, et nous réalisons l'analyse de Fourier. Les notations des parties précédentes ont été conservées.

3.2.1. *Préconditionnement diagonal en ordre 1*

Le système sans correction et sous-relaxation s'écrit alors :

$$(1 + \sigma \mid c \mid)\delta U^{n+1} = \widehat{\delta U} \tag{25}$$

avec $\widehat{\delta U}$ flux explicite d'ordre 1 (Partie I §2.4.).

Le schéma équivalent de (25) à l'ordre deux en espace est donné par :

$$U_t + c(1-\mu)cU_x = \frac{|c|\Delta x}{2}(1-\mu)U_{xx}$$

Il est clair que ce schéma est inconsistant : en effet la vitesse de l'onde est modifiée. Elle vaut $c(1-\mu)$ au lieu de c.

Soit après transformation de Fourier :

$$(1 + \sigma \mid c \mid)(g_k(\Delta t) - 1) = \sigma c^+ \exp(-i\xi_k) - \sigma \mid c \mid -\sigma c^- \exp(i\xi_k)$$

comme nous avons $1 + \sigma \mid c \mid > 0$, alors le facteur d'amplication vaut :

$$g_k(\Delta t) = \frac{1 + \sigma \mid c \mid \cos\xi_k - i\sigma c \sin\xi_k}{1 + \sigma \mid c \mid}$$

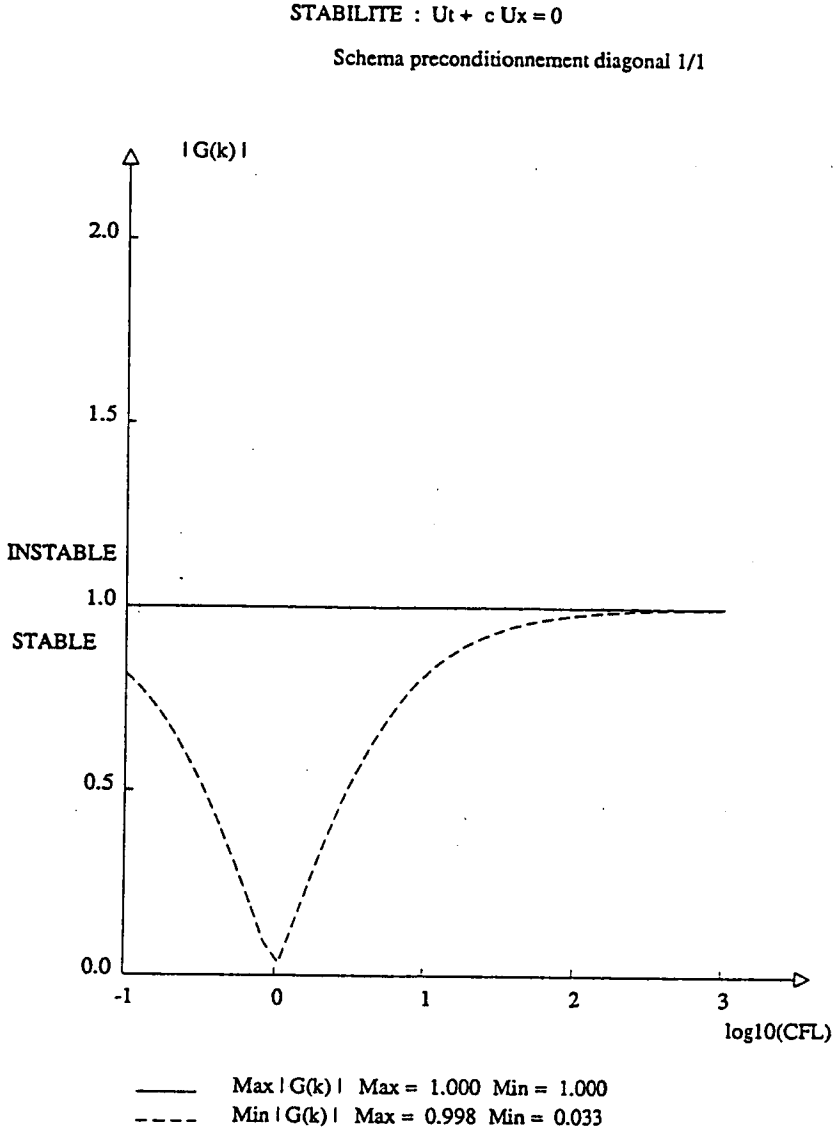


Fig. IV : *Stabilité du préconditionneur diagonal en ordre 1*

et :

$$| g_k(\Delta t) | = \sqrt{\frac{1 + \mu(2\cos\xi_k + \mu)}{1 + \mu(2 + \mu)}} \tag{26}$$

avec $\mu = \sigma \mid c \mid$.

Proposition 5 :

Le schéma préconditionnement diagonal ordre 1 est inconditionnellement l_2-stable pour l'équation de propagation.

En effet comme $\cos\xi_k \leq 1$ nous avons bien que $\mid g_k(\Delta t) \mid \leq 1$ pour tout $k \in [1, N]$ et $\mu \in I\!R^{*+}$.
$\oslash$

Lorsque le pas de temps est grand (pour μ tendant vers l'infini), nous constatons que le module du facteur d'amplification tend vers 1 quel que soit le mode de Fourier considéré, la convergence du schéma est donc ralentie par rapport à une méthode implicite. Le minimum du facteur d'amplification sur la courbe de la Figure **IV.** est obtenu sur une plage de pas de temps plus grande que pour un code explicite c'est à dire : $0,1 \leq \mu \leq 10$ ce qui permet d'espérer un gain d'efficacité.

3.2.2. *Phase physique d'ordre supérieur*

Nous considérons la méthode du préconditionnement diagonal avec un flux explicite d'ordre 2 (voir Partie II §3.1.2.) et le paramètre de sous-relaxation ω tel que $0 \leq \omega \leq 1$. Le calcul par Fourier donne :

$$(1 + \sigma \mid c \mid)\delta U_i^{n+1} = -\sigma\omega \left\{(1-\beta)c\delta^0 + \beta c^-\delta^+ + \beta c^+\delta^-\right\} \qquad (27)$$

avec :

$$\begin{cases} \delta^0 = \dfrac{U_{i+1}^n - U_{i-1}^n}{2} \\ \delta^+ = \dfrac{-U_{i+2}^n + 4U_{i+1}^n - 3U_i^n}{2} \\ \delta^- = \dfrac{3U_i^n - 4U_{i-1}^n + U_{i-2}^n}{2} \end{cases}$$

Le schéma équivalent à l'ordre deux s'écrit :

$$U_t + (\omega - \mu)cU_x = -\frac{\Delta x}{2}|c|\mu U_{xx}$$

Le schéma (27) est inconsistant.

Le facteur d'amplification s'écrit :

$$g_k(\Delta t) = \frac{1 + \sigma \mid c \mid (1 - \omega\beta(1-\cos\xi_k)^2) - i\sigma c\omega \sin\xi_k(1 + \beta(1-\cos\xi_k))}{1 + \sigma \mid c \mid}$$

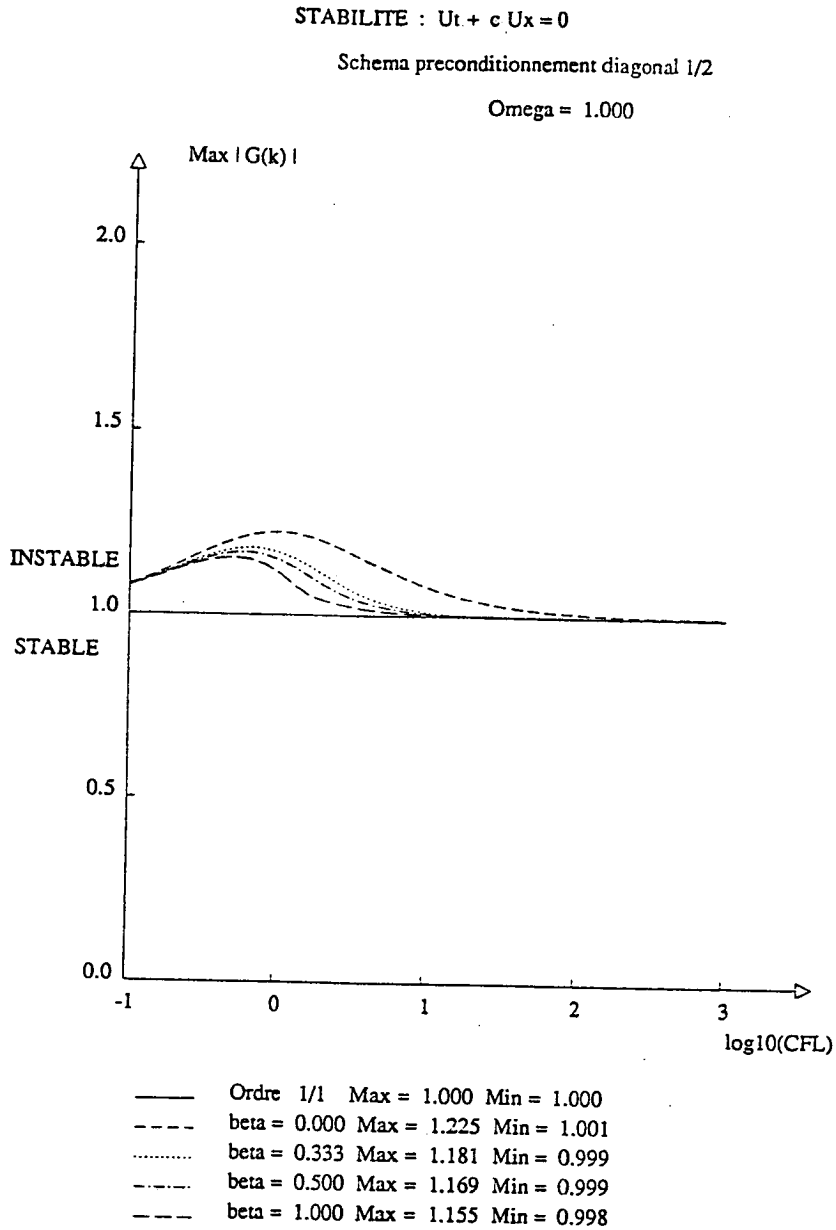


Fig. V : *a) Stabilité du préconditionneur diagonal en ordre 2*

L'étude du facteur d'amplification montre pour deux valeurs différentes de w (voir courbes de la Figure **VI. a) et b)**) que le schéma est **inconditionnellement**

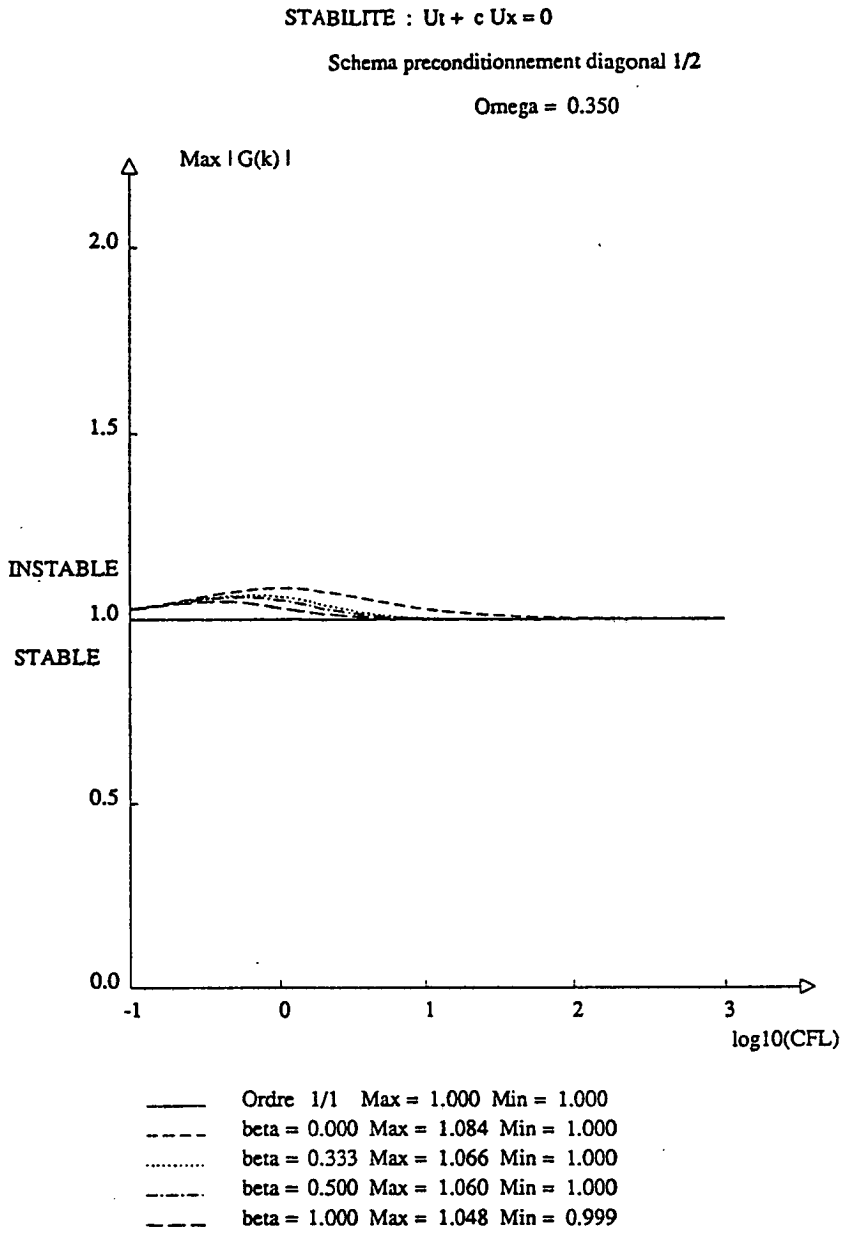


Fig. VI : *b) Stabilité du préconditionneur diagonal en ordre 2*

instable :

$$\max_{k\in[1,n]} \mid g_k(\Delta t) \mid \equiv 1 + \epsilon(\Delta t)$$

Où $\epsilon(\Delta t)$ est une fonction qui tend vers 0 lorsque Δt est grand quelques soient les valeurs de β et ω.

D'autre part les modes qui rendent instables le schémas sont **les basses fréquences** : ils sont d'autant moins nombreux à être instables que ω est petit, en particulier pour les cas où $0,25 \leq \omega \leq 0,5$. Mais il n'est pas possible de trop diminuer ω car la solution ne varie presque plus. Nous remarquons sur les différentes courbes présentées que $\omega = 0,35$ est une valeur proche de l'optimal au sens d'une meilleure stabilité surtout pour les grands pas de temps. Cette valeur est la valeur utilisée par A. Eberlé en [12].

De plus, comme dans le cas du schéma implicite d'ordre 2, les valeurs de β comprises entre 0,25 et 1 sont les valeurs qui réalisent la meilleure stabilité : nous trouvons pour $\epsilon(\Delta t)$ une valeur maximum proche de 0,05 avec $\omega = 0,35$ pour des CFL de l'ordre de 1 (donc l'instabilité est rendue très légère).

Le cas particulier $\beta = 0$ est un cas très *pathologique* puisque que pour tout mode de Fourier le module du facteur d'amplication est plus grand que 1. Par contre, le cas $\beta = 1$ est optimal.

Enfin, nous constatons que contrairement au cas du préconditionnement diagonal d'ordre 1, il est plus avantageux d'utiliser des grands pas de temps pour obtenir une meilleure convergence.

4. Tests numériques

Coûts et efficacités relatives : Nous avons comparé les divers algorithmes étudiés précédemment ; c'est à dire le schéma implicite avec Gauss-Seidel que l'on nommera par la suite schéma GS, le schéma implicite sans stockage matriciel avec Jacobi nommé J, le schéma préconditionneur diagonal nommé SD et le schéma explicite appellé EXP. Nous remarquons que le *gel* de la matrice est possible pour le schéma GS ce qui améliore l'efficacité mais, dans des problèmes difficiles tels les régimes à grand nombre de Mach il n'est pas possible de geler la matrice, ni d'accroître le CFL pendant les premières itérations car les variations sont trop grandes d'un instant à l'autre et les instablités apparaissent. C'est pourquoi nos premières comparaisons concernent un cas difficile avec des petits pas de temps tout d'abord. Le tableau (Figure 7 en fin de rapport) concerne le cas d'un écoulement à $Mach_\infty = 8$ autour d'un corps émoussé (697 nœuds) en ordre 1 et 2.

Nous présentons les coubes de convergence associées sur les Courbes 8.1 et 8.2.

Lorsque le nombre d'itérations linéaires est faible (de l'ordre de la dizaine), le schéma J reste efficace par rapport au schéma EXP : le facteur calculé est alors de 3. Par contre ce facteur d'efficacité décroît d'autant plus que le nombre d'itérations de Jacobi est grand. En effet le coût d'une itération de Jacobi est équivalent au coût de calcul d'un simple flux implicite d'ordre 1.

Quand il possible d'utiliser des grands CFL dès les premières itérations (non linéaires) alors il est peut être intéressant de résoudre presque complètement le système linéaire en augmentant le nombre de relaxations. Dans ce cas, la méthode de Jacobi avec stockage reste compétitive avec GS.

Les comparaisons entre les différents schémas implicites *en ordre deux* mènent aux mêmes conclusions qu'en ordre un. Pour le cas du schéma avec préconditionnement diagonal, la convergence est difficile : (voir la courbe 8). Des anomalies apparaissent dues à des instabilités et des singularités dans le préconditionneur. Lorsque le paramètre de relaxation ω passe de 1 à 0,35 la convergence est améliorée (le cas $\omega = 1$ diverge) ce qui avait été prédit par la théorie dans un cas simplifié (voir le paragraphe précédent). Lorsque $\omega = 0,35$, nous obtenons une solution quasi-stationnaire après 300 itérations (résidu à 10^{-4}) qui semble évoluer vers un comportement divergent, ce qui s'interprête par la croissance des modes basses fréquences prédite par l'analyse de Fourier.

5. Conclusion

On a construit une méthode implicite linéarisée qui évite un encombrement mémoire supplémentaire dû à la matrice, en particulier lorsque les dérivées approchées utilisées pour le calculs des termes d'ordre 2 sont déjà stockées ; mais les résultats ont été obtenus au prix d'une moins bonne efficacité, qui reste cependant très importante devant un algorithme explicite.

D'autre part nous avons réussi à maximiser l'efficacité dans le cas de certains problèmes : tout d'abord lorsque le CFL est grand dès le dèbut de l'évolution, un grand nombre d'itérations linéaires compense le fait d'utiliser de grands pas de temps ; ensuite lorsque le problème est difficile à résoudre et que l'on ne peut pas utiliser de grands CFL, alors peu d'itérations linéaires s'impose.

Il est maintenant intéressant de présenter l'application d'un code sans stockage sur un calculateur vectoriel.

IV. Adaptation au calcul vectoriel

Dans la perspective d'une efficacité maximale du programme implicite linéarisé nous nous sommes intéressés au problème de l'utilisation d'un code de ce type sur un calculateur vectoriel. Les calculs ont été réalisés sur le CRAY-1S et CRAY-2 à partir du frontal du C.C.V.R. à Palaiseau.

Nous avons réalisé une version à faible stockage matriciel, avec les flux de Osher pour la phase physique et de Steger-Warming pour la phase mathématique (voir les parties précédentes pour la construction des flux). Cette version fait suite à une étude réalisée par F. Angrand et J. Erhel [14] à partir de [4] avec une version implicite avec stockage matricielle combinant les flux de Vijayasundaram [5] (phase mathématique) et le Q-Schéma dérivé d'un schéma proposé par Hancock et van Leer [3] (phase physique).

1. Organisation du calcul vectoriel

Nous renvoyons aux parties précédentes la description des méthodes de résolution. Nous examinons plus en détail la vectorisation des flux et la résolution du système linéaire.

1.1. *Traitement des boucles sur les segments et addressage indirect*

Nous étudions tout d'abord la phase physique. D'une manière général, il s'agit de calculer en chaque cellule C_i le terme $\widehat{\delta W_i}$:

$$\widehat{\delta W_i} = -\frac{\Delta t}{mes(C_i)} \left\{ \sum_{j \in K(i)} \Phi_{ij}^n \; + \; \text{termes de bord} \right\}$$

Mais en réalité, nous ne connaissons pas explicitement l'ensemble des nœuds voisins à i : $K(i)$. Le calcul des flux est réalisé par segments sur les interfaces des cellules.

Les boucles sur les interfaces (entre deux cellules) utilisent des variables stockées par nœuds, par exemple les solutions W_i^n, nous devons alors utiliser des adressages indirects entre les différentes variables. On utilise les opérations SCATTER et GATHER qui correspondent aux deux différents types d'adressages.

L'opération *GATHER* se met sous la forme : $Y(I) = X(INDEX(I))$ pour I variant de $I1$ à $I2$. Elle est toujours vectorisable, elle réalise le transfert des tableaux élémentaires (les nœuds) à des tableaux sur lesquels les boucles sont vectorisables (par exemple les segments).

L'opération SCATTER est définie par : $X(INDEX(I)) = Y(I)$ pour I variant de $I1$ à $I2$. Elle accumule des résultats élémentaires (par exemple les nœuds). Cette opération est vectorisable si $INDEX(I)$ est une fonction injective de la variable I.

Dans le cas d'une boucle par segments $[a_i, a_j]$, nous calculons les états W_{ij}^n et W_{ji}^n à l'aide d'un GATHER, puis nous évaluons le flux $\Phi_{ij}^n = \Phi(W_{ij}^n, W_{ji}^n)$. Enfin nous assemblons les flux pour les sommets a_i et a_j par un SCATTER. Pour chaque segment numéroté ij nous calculons :

$$\begin{cases} \widehat{\delta W_i} = \widehat{\delta W_i} - \dfrac{\Delta t}{mes(C_i)} \Phi_{ij}^n \\ \widehat{\delta W_j} = \widehat{\delta W_j} + \dfrac{\Delta t}{mes(C_j)} \Phi_{ij}^n \end{cases} \tag{28}$$

La vectorisation de la boucle sur les segments n'est possible que si les données sont indépendantes à l'interieur de la boucle. Or, ici ce n'est pas toujours vrai : en effet chaque calcul par segment nécessite l'utilisation des variables sur les deux extrémités (valeurs à droite et à gauche du flux), or il se peut que au moins deux segments utilisent dans la boucle, des valeurs aux extrêmités communes (en effet un nœud quelconque possède toujours plusieurs nœuds voisins). En conséquence, si on force la vectorisation de la boucle (28), les valeurs de $\widehat{\delta W_i}$ peuvent être *écrasées* (c'est à dire non prises en compte) pour deux segments d'un même registre vectoriel qui ont comme extrémité commune a_i.

Pour remédier à cet inconvénient, F. Angrand et J. Erhel [14] ont eu recours à un algorithme de **coloriage** décrit en [24]. Cet algorithme construit une partition des segments telle que ceux-ci soient totalement indépendants entre eux. Pour des raisons d'efficacité, on cherche à minimiser le nombre total de couleurs (ou des parties de la partition); le calcul vectoriel nécessite des vecteurs de taille suffisamment grande. Dans l'algorithme proposé le nombre de couleurs minimal est borné par la formule suivante :

$$\max_{1 \le i \le NS} d_i \le NC \le 3 * \max_{1 \le i \le NS} d_i - 5$$

Où d_i est le nombre de nœuds voisins de i dans la triangulation. NC est le nombre de couleurs et NS le nombre de nœuds total. Ainsi pour $d_i = 6$ (triangulation régulière avec un maillage à 7 points), $6 \le NC \le 13$ (voir la Fig. VII).

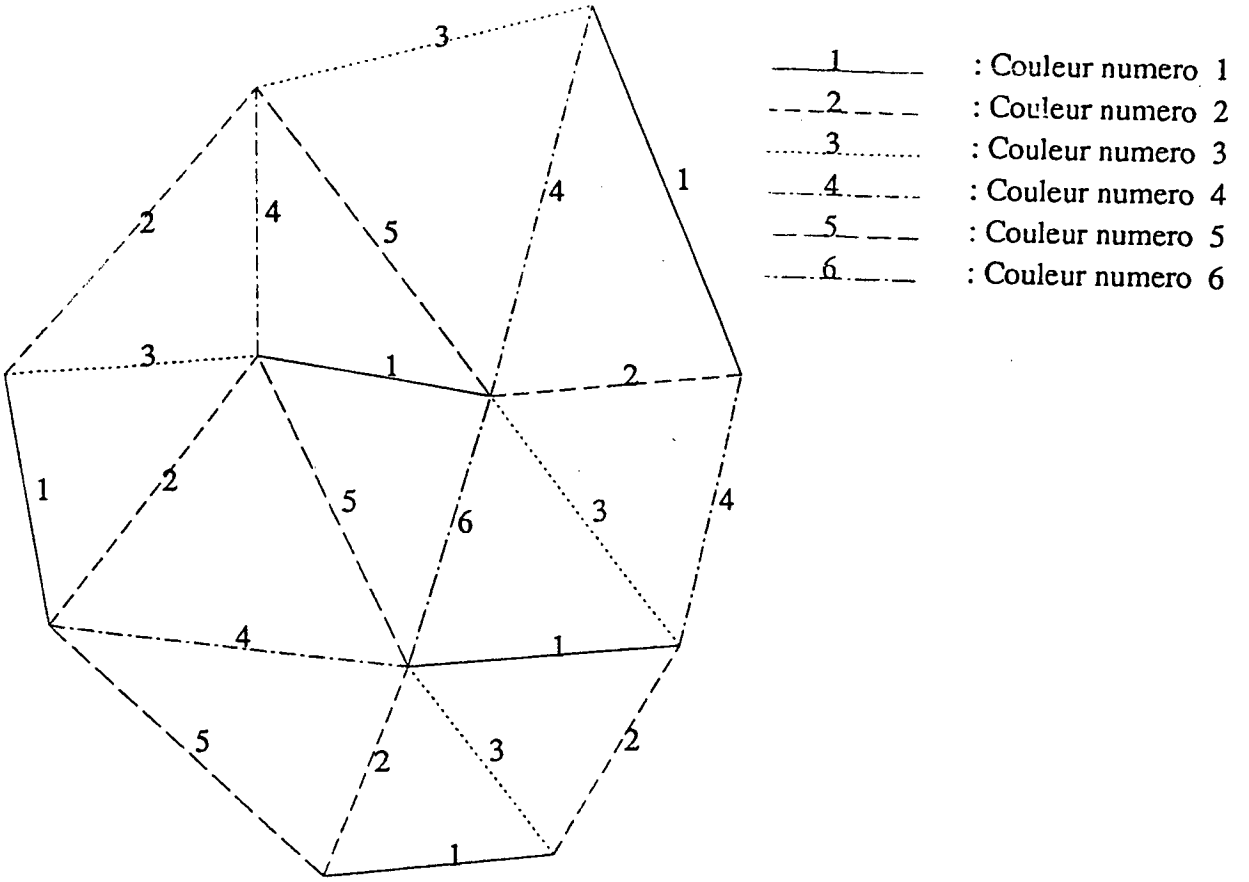


Fig. VII : *Exemple de coloriage sur les segments*

Description de l'algortithme de partition en couleurs :

– On a NS nombre total de nœuds du maillage; $NSEG$ nombre total de segments et $NCAMAX$ nombre maximal de couleur autorisé.

a) Initialisations :
On introduit des marqueurs sur les nœuds : $MARK(1 : NS)$. Avec $MARK(IS) = 1$ si le nœud IS appartient déjà à un segment colorié et $MARK(IS) = 0$ sinon. Le tableau $MARK$ est mis à 0.
On introduit aussi le tableau $COL(1 : NSEG)$ qui indique la couleur de chaque segment.
Enfin on construit le tableau $ICOLA(1 : NCAMAX)$ qui indique le numéro du dernier segment colorié pour cette couleur.
On pose $NC = 0$, $NCOL = 0$ et $ICOLA$ est mis à 0. NC est le nombre de couleur déjà utilisé et $NCOL$ nombre de segments déjà coloriés.

b) Boucle sur les couleurs :

Si le test $NC = NCAMAX$ est vrai alors le nombre maximal de couleurs est atteint donc il faut augmenter $NCAMAX$.
Si le test $NCOL < NSEG$ est vrai alors il y a encore des segments non coloriés donc on change la couleur : $NC = NC + 1$; sinon l'algorithme est terminé.

— Boucle sur les tous les segments :
Pour chaque segment $ISEG$ on a les deux extrêmités $IS1$ et $IS2$. Si le test $COL(ISEG) \neq 0$ est vrai alors le segment est déjà colorié, on passe au suivant.
Si le test $MARK(IS1) = 1$ OU $MARK(IS2) = 1$ est vrai alors on ne colorie pas ce segment : la couleur courante NC ne convient pas et on passe au segment suivant ; sinon on colorie le segment $ISEG$ par la couleur NC en posant :

$$\begin{cases} COL(ISEG) = NC \\ MARK(IS1) = 1 \ ; \ MARK(IS2) = 1 \\ NCOL = NCOL + 1 \end{cases}$$

Si la boucle sur les segments est terminée, on passe à la couleur suivante (boucle sur les couleurs).

c) Renumérotation des segments :
La renumérotation est réalisée de telle façon à ranger les segments par couleur d'un manière croissante.

d) Fin de L'algorithme :
NC est le nombre total de couleur, $ICOLA$ contient pour chaque couleur le numéro du dernier segment.

Remarque :

L'algorithme décrit plus haut n'est pas optimal. En effet, lorsque les maillages ne sont pas réguliers, il crée des vecteurs de longueur trop courts; ce qui ralentit les performances de la vectorisation. Néanmoins, plus de 90% des segments sont dans des vecteurs de longueur suffisante (c'est à dire supérieure à la longueur du registre vectoriel).

1.2. *Vectorisation du flux de Osher*

Description [7] :

Le flux de Osher fait partie des méthodes de résolution approchée du problème de Riemann pour un système hyperbolique. Nous notons par U_G et U_D les valeurs à gauches et à droites du problème de Riemann. Le flux de Osher se construit

naturellement comme une méthode de décomposition de flux *plus-moins*, ici en *Difference-Flux-Splitting* :

$$\begin{aligned}\Phi(U_G,U_D) &= F^+(U_G) + F^-(U_D) \\ &= F(U_G) + (F^-(U_D) - F^-(U_G)) = F(U_G) + \int_{U_G}^{U_D} A^-(W)dW \\ &= F(U_D) - (F^+(U_D) - F^+(U_G)) = F(U_D) - \int_{U_G}^{U_D} A^+(W)dW \\ &= \frac{1}{2}\left(F(U_G) + F(U_D) - \int_{U_G}^{U_D} |A(W)|dW\right)\end{aligned}$$

où $F(W)$ est la première composante du flux continu d'Euler et $A(W)$ son Jacobien.

Il s'agit alors de déterminer le domaine d'intégration des intégrales :

$$\int_{U_G}^{U_D} A^+(W)dW \ , \ \int_{U_G}^{U_D} A^-(W)dW$$

Supposons que les deux états U_G et U_D sont liés entre eux sur le domaine d'intégration noté Γ_k qui est tangent au vecteur propre R_k de la matrice $A(W)$:

$$\frac{dW}{d\xi}(\xi) = R_k(W(\xi)) \ , \ U_G = W(O) \ , \ U_D = W(\xi_D)$$

où ξ est une paramétrisation de la courbe Γ_k.

On effectue alors le changement de variable sur l'intégrale :

$$\begin{aligned}\int_{U_G}^{U_D} A^-(W)dW &= \int_0^{\xi_R} A^-(W(\xi))\frac{dW}{d\xi}d\xi \\ &= \int_0^{\xi_R} A^-(W(\xi))R_k(W(\xi))d\xi \\ &= \int_0^{\xi_R} \lambda_k^-(W(\xi))R_k(W(\xi))d\xi\end{aligned}$$

pour λ_k k^{ieme} valeur propre associée associée au vecteur propre R_k.

Lorsque cette valeur propre est positive sur $\Gamma_k = [0, \xi_R]$ l'intégrale est nulle, si elle est négative l'intégrale vaut $F(U_D) - F(U_G)$, enfin si elle change de signe il existe $\xi = \xi^s \in]0, \xi_R[$ où λ_k=0. Posons $U^s = W(\xi^s)$, ce point est appelé *point sonique*.

Si la valeur propre est positive (respectivement négative) pour $\xi < \xi^s$ et négative (respectivement positive) pour $\xi > \xi^s$ alors l'intégrale vaut $F(U_D) - F(U^s)$ (respectivement $F(U^s) - F(U_G)$).

Par ailleurs, nous avons les résultats suivant : si le vecteur propre $R_k(W)$ est vraiment non linéaire (ou VNL) alors la valeur propre associée λ_k change de signe au plus une seule fois sur Γ_k ; par contre si $R_k(W)$ est linéairement dégénéré (ou LD) alors λ_k reste constant sur Γ_k.

Soit $W = (\rho, u, v, p)$ et $\tilde{u} = u\mu_1 + v\mu_2$, $(\mu_1, \mu_2) \in I\!R^2$. Dans le cas des équations d'Euler 2-D, les deux valeurs propres $\lambda_2 = \lambda_3 = \tilde{u}$ qui correspondent au cas LD et les deux autres valeurs propres $\lambda_1 = \tilde{u} + c$; $\lambda_4 = \tilde{u} - c$ pour le cas VNL. Ainsi, il peut apparaître deux points soniques : le premier noté U_D^s se trouve alors sur Γ_1 et le second noté U_G^s sur Γ_4.

Le calcul de flux peut se décomposer de la manière suivante :

$$\begin{aligned}\Phi(U_G, U_D) &= F(U_G) + \int_{U_G}^{U_D} A^-(W)dW \\ &= F(U_G) + \left(\overbrace{\int_{U_G}^{U^{1/3}}}^{\Gamma_1} + \overbrace{\int_{U^{1/3}}^{U^{2/3}}}^{\Gamma_2 \cup \Gamma_3} + \overbrace{\int_{U^{2/3}}^{U_D}}^{\Gamma_4} \right) A^-(W)dW\end{aligned}$$

avec $U^{1/3}$ et $U^{2/3}$ sont les deux états intermédiaires qui sont calculés grâce aux invariants de Riemann (connus pour Euler), et, qui sont constants le long de chaque domaine d'intégration Γ_k pour $k = 1, .., 4$. De même, les points soniques sont évalués par la connaissance des invariants.

Nous notons par $\lambda_k^G, \lambda_k^{1/3}, \lambda_k^{2/3}, \lambda_k^D$, les valeurs propres des quatres états *à gauche*, intermédiaires, et *à droite*; et par $c^{1/3}, c^{2/3}$ les vitesses du son des états intermédiaires.

Au total, les tests à faire sont résumés dans le tableau suivant :

	$\lambda_1^G > 0$ *et* $\lambda_4^D < 0$
$0 < \lambda_4^{1/3}$	$F(U_G) - F(U_G^s) + F(U_D)$
$-c^{1/3} < \lambda_4^{1/3} < 0$	$F(U_G) - F(U^{2/3}) + F(U_D)$
$0 < \lambda_1^{1/3} < c^{1/3}$	$F(U_G) - F(U^{1/3}) + F(U_D)$
$\lambda_1^{1/3} < 0$	$F(U_G) - F(U_D^s) + F(U_D)$

	$\lambda_1^G > 0$ et $\lambda_4^D > 0$
$0 < \lambda_4^{1/3}$	$F(U_G)$
$-c^{1/3} < \lambda_4^{1/3} < 0$	$F(U_D^s) - F(U^{2/3}) + F(U_G^s)$
$0 < \lambda_1^{1/3} < c^{1/3}$	$F(U_G) - F(U^{1/3}) + F(U_G^s)$
$\lambda_1^{1/3} < 0$	$F(U_G) - F(U_D^s) + F(U_G^s)$

	$\lambda_1^G < 0$ et $\lambda_4^D < 0$
$0 < \lambda_4^{1/3}$	$F(U_D^s) - F(U_G^s) + F(U_D)$
$-c^{1/3} < \lambda_4^{1/3} < 0$	$F(U_D^s) - F(U^{2/3}) + F(U_D)$
$0 < \lambda_1^{1/3} < c^{1/3}$	$F(U_D^s) - F(U^{1/3}) + F(U_D)$
$\lambda_1^{1/3} < 0$	$F(U_D)$

	$\lambda_1^G < 0$ et $\lambda_4^D > 0$
$0 < \lambda_4^{1/3}$	$F(U_D^s)$
$-c^{1/3} < \lambda_4^{1/3} < 0$	$F(U_D^s) - F(U^{2/3}) + F(U_D)$
$0 < \lambda_1^{1/3} < c^{1/3}$	$F(U_D^s) - F(U^{1/3}) + F(U_G^s)$
$\lambda_1^{1/3} < 0$	$F(U_G^s)$

La remarque principale concernant la vectorisation de ce flux est le grand nombre de tests nécessaires. Les tests classiques inhibent la vectorisation sur le Cray-1S ce qui nous conduit à appliquer un traitement particulier décrit au paragraphe suivant.

Calcul vectoriel pour le flux de Osher :

Nous avons assemblé le flux de Osher à partir de la version fonctionnant en mode *scalaire* décrite par S. Osher et F. Solomon [7] et écrite en éléments finis par B. Stoufflet - L. Fezoui [31]. Nous décrivons le flux Osher en mode *vectoriel* :

Nous itérons sur les segments pour chaque couleur pour assurer la vectorisation de l'opération SCATTER, et, par paquets de longueur $LVECT$ pour optimiser le calcul vectoriel et pour éviter le stockage d'un nombre considérable de tableaux intermédiaires pour le calcul des flux. $LVECT = 64$ semble un bon choix sur Cray-1S car il correspond au stockage maximal d'un tableau dans un registre vectoriel.

a) Nous opérons un GATHER des variables stockées par nœuds :

$$\begin{cases} WS1(ISEG) = W(IS1) \\ WS2(ISEG) = W(IS2) \end{cases}$$

où $W(IS1)$ est la variable d'état à gauche (valeur interpolée au nœud de numéro $IS1$) avec $IS1 = INDEX1(ISEG)$ et $W(IS2)$ est la valeur à droite (valeur interpolée au nœud de numéro $IS2$) avec $IS2 = INDEX2(ISEG)$.

b) Puis nous calculons le flux de Osher :

$$\Phi(WS1(ISEG), WS2(ISEG))$$

Les boucles internes en $ISEG$ doivent être vectorisables : dans la version en mode scalaire, il y a des tests locaux concernant l'apparition de points soniques pour le calcul des quantités exprimant des conditions vraiment non linéaires (VNL) et sur le signe des valeurs propres pour déterminer le sens de décentrage (voir le paragraphe concernant la description du flux de Osher). Les tests dépendent de $ISEG$, le CRAY-1S inhibe la vectorisation des boucles lors de leurs apparitions. C'est pourquoi, pour les éviter, nous introduisons une variable logique locale supplémentaire qui contient le résultat du test. Puis nous calculons les valeurs cherchées en tenant compte de cette variable.

Exemple : Soient F la variable à calculer et $\mathcal{T}$ la variable logique test, pour chaque valeur de i de $i1$ à $i2$ on a :

$$\begin{cases} F(i) = F1(i) & \text{si } \mathcal{T}(i) \text{ est vrai} \\ F(i) = F2(i) & \text{si } \mathcal{T}(i) \text{ est faux} \end{cases}$$

Nous remplaçons la formule ci-dessus non vectorisable par une autre vectorisable en précalculant une variable $T(i)$ (par une formule arithmétique) qui vaut 1 quand $\mathcal{T}(i)$ est vrai et 0 sinon. Nous obtenons alors pour $F(i)$:

$$F(i) = F1(i)T(i) + F2(i)(1 - T(i))$$

Et il est clair que les deux formulations de $F(i)$ sont équivalentes pour tout i mais il y a un surplus de calcul par rapport à la version contenant : nous avons doublé le calcul.

c) Enfin, l'opération SCATTER s'écrit :

$$\begin{cases} DW(IS1) = DW(IS1) - \Phi(WS1(ISEG), WS2(ISEG)) \\ DW(IS2) = DW(IS2) + \Phi(WS1(ISEG), WS2(ISEG)) \end{cases}$$

Les variables DW dénotent le flux partiel. Avec $IS1 = INDEX1(ISEG)$, $IS2 = INDEX2(ISEG)$, les fonctions $INDEX1$ et $INDEX2$ qui distribuent les deux nœuds de chaque segment sont bien sûr injectives grâce au coloriage effectué en amont du programme.

1.3. *Phase mathématique*

En premier lieu, l'assemblage des termes diagonaux de la matrice implicite se fait de manière analogue à celui de la phase physique : les contributions des flux

sont remplacées par des contributions des flux linéarisés avec la décomposition de Steger-Warming. Il n'y a pas de tests locaux contrairement au flux de Osher.

Le système linéaire est résolu par la méthode de Jacobi sans le stockage des termes extra-diagonaux : à chaque itération linéaire, nous calculons l'itéré X^{iter+1} en fonction de l'itéré X^{iter} de la manière suivante :

A chaque couleur des segments (boucle par couleur de numéro NC) nous réalisons une boucle sur des paquets de vecteurs de longueur maximale $LVECT$ ($LVECT$ = est un multiple de 64 ici pour optimiser le calcul vectoriel).

a) Nous faisons un GATHER des variables stockées par nœuds : W^n, X^{iter} en des varaibles stockées par segments.

b) Puis nous calculons le second membre de Jacobi :

$$S = \widehat{\delta W} - E^n X^{iter}$$

Le terme $\widehat{\delta W}$ représente le flux explicite, et E^n sont les termes extra-diagonaux calculés sur les segments.

c) Nous calculons par nœuds la solution à l'itération $iter + 1$:

$$X^{iter+1} = (D^n)^{-1} S$$

D^n étant la diagonale stockée par nœud de la matrice implicite.

d) Enfin l'opération GATHER assemble le tableau X^{iter+1} sur les nœuds.

La partie b) est trivialement vectorisable : elle contient près de 80% du calcul des flux.

2. Tests numériques

L'adaptation du calcul en mode vectoriel augmente sensiblement les performances par rapport au calcul en mode scalaire. Sur CRAY-1S, la comparaison du même code en mode scalaire et en mode vectoriel montre une accélération d'un facteur 4.

Le schéma implicite avec stockage complet de la matrice limitait à environ 3000 nœuds la taille maximale du maillage sur CRAY-1S, tandis le non stockage des termes non diagonaux permet maintenant des maillages de 10000 nœuds soit au

moins 3 fois plus qu'avant. Ceci nous laisse espérer l'utilisation de maillages plus fins et complexes sur le CRAY-2, cette machine ayant une place mémoire plus grande (256 Mmots contre 1 Mmots pour le CRAY-1S) ce qui permet de faire tourner en mémoire centrale un code tridimensionnel implicite *sans stockage matriciel* avec des maillages non triviaux (> 50000 points par exemple).

Nous présentons des calculs sur le CRAY-1S avec un tableau de comparaisons avec les différentes versions implicites en mode scalaire et en mode vectoriel :

Test sur NACA0012 avec 800 points :

Nous présentons deux calculs sur CRAY-1S des codes *vectoriels* implicites linéarisés avec stockage matriciel et sans stockage matriciel. Les tableaux 1,1 bis, 1 ter représentent le coût moyen par itération non linéaire des différents sous-programmes du code :
– Le sous-programme noté *relax* : résolution du système linéaire par la méthode de Jacobi par segments.
– Le sous-progamme noté *calmat* : assemblage de la matrice entière pour la méthode avec stockage et assemblage des blocs diagonaux pour la méthode sans stockage.
– Le sous-programme noté *osher* : calcul du flux Osher.
– Les sous-programmes restant notés *divers* : calcul du pas de temps et des gradients, conditions au bord, inversion des blocs diagonaux ...

Tableau 1 : Coût des méthodes implicites sur CRAY-1S

nbrel= 4	Jacobi	sans stockage	Jacobi	avec stockage
(*)	CPU sec	%	CPU sec	%
relax	0.0963	40.8	0.0406	22.2
calmat	0.0490	21.4	0.0562	30.7
osher	0.0590	25.8	0.0590	32.2
divers	0.0275	12.0	0.0273	14.9
Total	0.2291	100.0	0.1831	100.0

Tableau 1 bis

nbrel=50	Jacobi	sans stockage	Jacobi	avec stockage
(*)	CPU sec	%	CPU sec	%
relax	1.1700	89.6	0.5075	78.0
calmat	0.0489	3.7	0.0565	8.7
osher	0.0592	4.5	0.0589	9.1
divers	0.0275	2.1	0.0276	4.2
Total	1.3066	100.0	0.6505	100.0

Tableau 1 ter : Rapports de coût de l'implicite
avec et sans stockage

avec stockage / sans stockage	Scalaire (**)	Vectoriel (***)
nbrel= 4	1.33	1.25
nbrel=50	3.00	2.00

(*) *nbrel* est le nombre d'itérations linéaires.
(**) Voir chapitre II. : la méthode avec stockage matriciel utilise la résolution par Gauss-Seidel avec $nbrel = 2$.
(***) Voir tableaux précédents.

Rapports Vectoriel-Scalaire sur Cray-1S et Cray-2 :

Le rapport Vectoriel-Scalaire sur Cray-1S est d'environ 4 sur le code implicite linéarisé. Ce résultat ne change pas avec le code sans stockage matriciel.

De plus des calculs plus récents sur Cray-2 montrent un gain d'environ 2, 5 par rapport au Cray-1S. La phase de *compilation* sur Cray-2 étant effectuée avec l'option *cft77* (fortran 77) alors qu'elle était en fortran 66 sur Cray-1S. D'autre part, ce gain s'explicite par un meilleur rapport d'efficacité des opérations SCATTER et GATHER lors de la résolution du système linéaire.

Nous donnons les temps CPU par itération et par points des méthodes implicites par rapport à un calcul explicite. Les tests sont effectués sur Cray-2 avec le NACA0012 en ordre 2 avec CFL= numéro de l'itération et le nombre d'itérations linéaires est 4.

Tableau 2 : Coût sur Cray-2 des codes de calcul

Codes de calcul	Coût (*)	ratio
Explicite CFL=0.6	43.	1.0
Implicite avec stockage	152.	3.6
Implicite sans stockage	193.	4.5

(*) en microsecondes par itération et par point.

Les rapports d'efficacité pour atteindre la convergence vers la solution stationnaire (résidu inférieur à 10^{-3}) sont données par le tableau 3 :

Tableau 3

Codes de calcul	Nombre iter	CPU en sec.	ratio
Explicite CFL=0.6	600	20.6	1.0
Implicite avec stockage	40	5.0	4.1
Implicite sans stockage	40	6.1	3.4

Remarque :

Dans le cas transsonique, le gel de la matrice sur quelques itérations est possible avec le code implicite avec stockage complet, ce qui augmenterait encore l'efficacité de ce code devant les autres.

V. Conclusion

Les schémas implicites linéarisés sont particulièrement efficaces lorsque l'on peut utiliser des grands pas de temps ($CFL \geq 1000$) même si le nombre de balayages dans la relaxation est important (proche de 15 pour Gauss-Seidel, de l'ordre de 30 pour Jacobi : mais ce nombre est dépendant de la discrétisation).

Pour des écoulements supersoniques avec des nombres de Mach plus grand que 5, la phase de recherche de la solution reste assez coûteuse mais l'utilisation d'algorithmes plus sophistiqués, tels notamment les méthodes de sous-domaines, devraient augmenter encore les performances. Par ailleurs, la comparaison avec des codes explicites montre que le code implicite est plus robuste (surtout en ordre 2).

L'utilisation d'un calculateur vectoriel, nous a permis d'accroître les performances du code implicite linéarisé avec faible encombrement mémoire. Avec une taille mémoire proche de celle du code explicite, l'algorithme implicite à faible stockage matriciel s'est avéré compétitif, y compris pour des écoulements à grand nombre de Mach lorsqu'il n'est pas possible d'utiliser des grands CFL lors de la phase de recherche de la solution. Par ailleurs, dans la phase de convergence, les grands CFL sont possibles sans qu'il soit nécessaire de résoudre complétement le système linéaire.

Dans une étude ultérieure, nous envisageons l'utilisation d'une décomposition de flux de van-Leer [21] pour les phases explicite et mathématique. Ce flux a déjà donné de bons résultats à grand nombre de Mach avec un code explicite ([20]); il a été aussi implémenté pour un calcul implicite dans [19].

Aussi, l'application de la méthodogie des multigrilles à un code implicite linéarisé à faible stockage matriciel est en cours d'étude, elle devrait permettre d'augmenter encore les performances compte tenu des performances déjà atteintes du code multigrille explicite (voir [15], [27]).

Enfin, les applications essentielles d'un code implicite sans-stockage sont particulièrement avantageuses là où les calculs demandent un grand nombre de nœuds, principalement pour des écoulements tridimensionnels : le stockage de la matrice implicite est trois plus important en 3-D qu'en 2-D.

Figure 1 :
Géométrie : Canal du GAMM avec dos d'âne
hauteur maximale 4,2%
Nombre de points = 1512

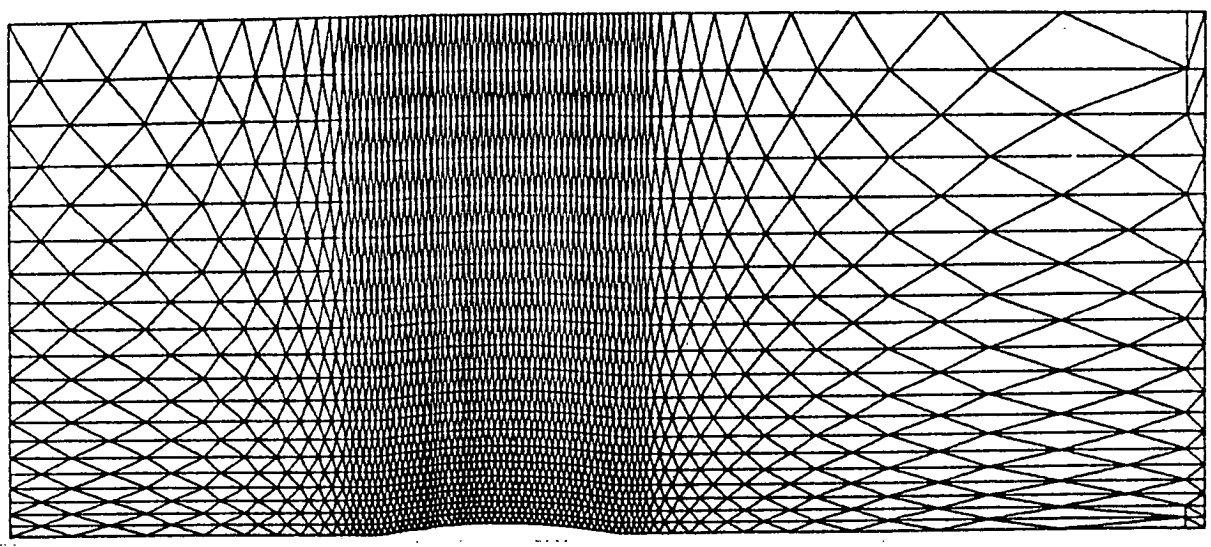


Figure 2.1 :
Courbe de convergence : canal du GAMM

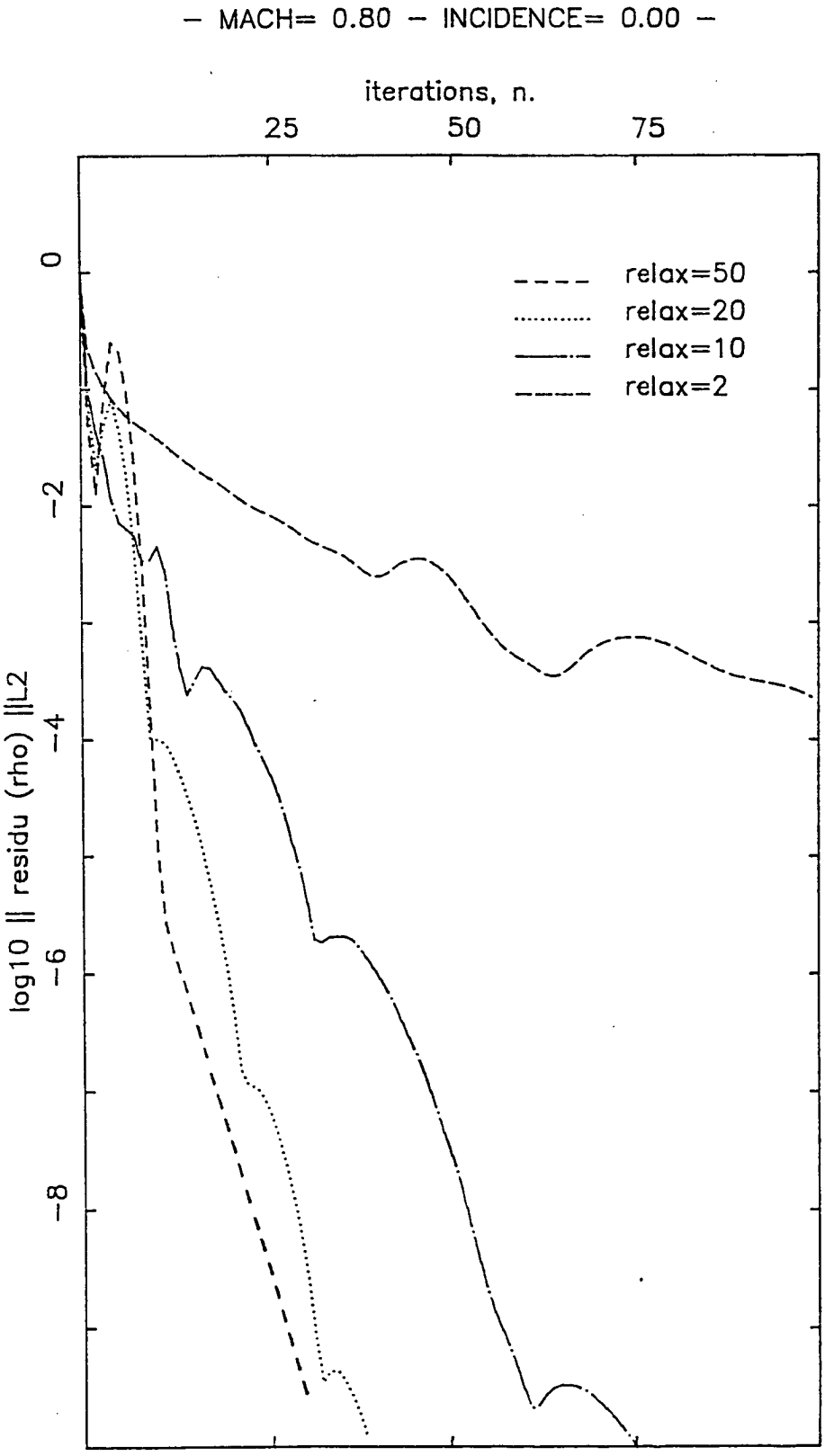


Figure 2.2 :
Courbe de convergence : canal du GAMM

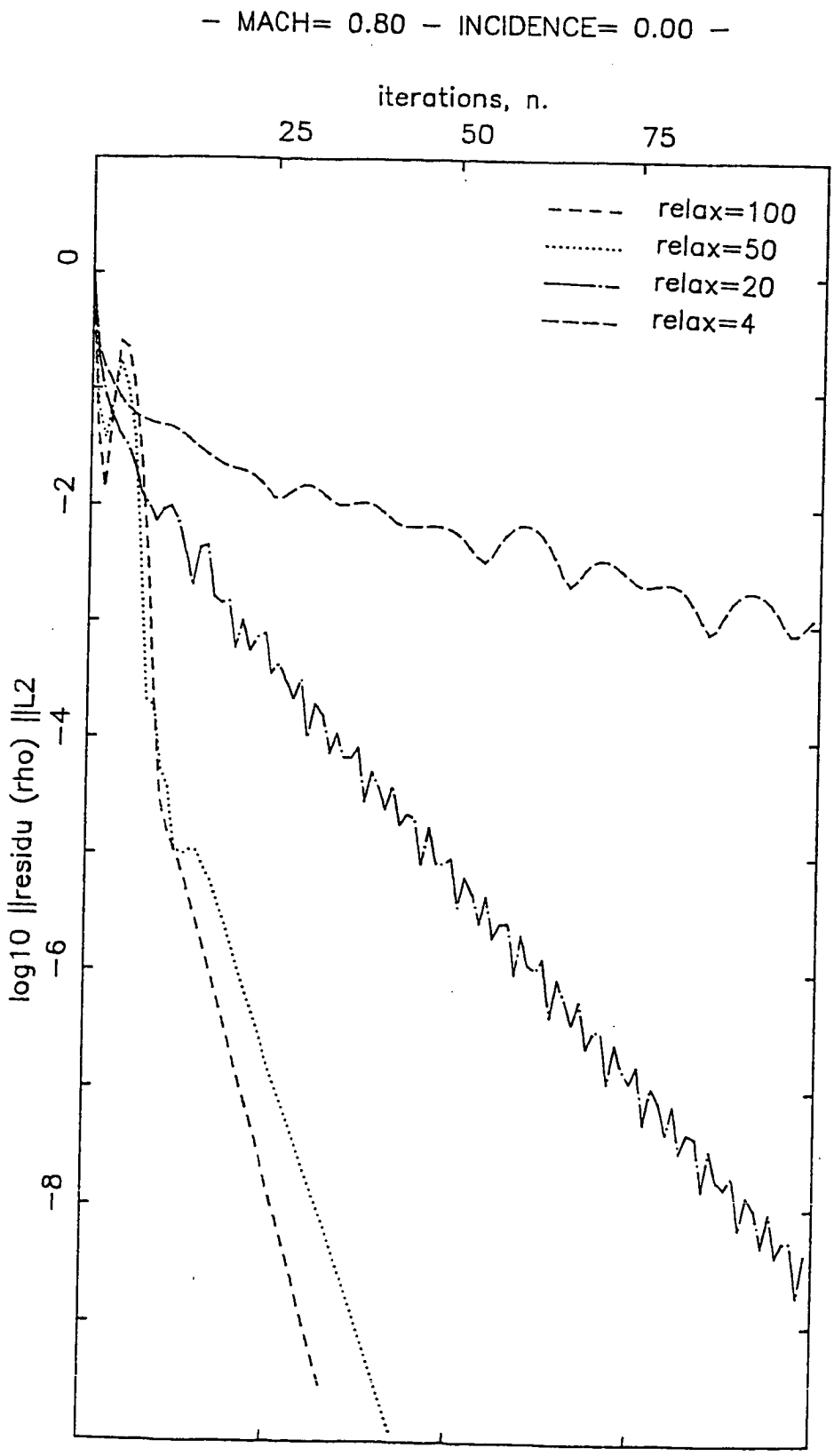


Figure 2.3 :

COMPARAISON D'EFFICACITE ENTRE
GAUSS-SEIDEL ET JACOBI

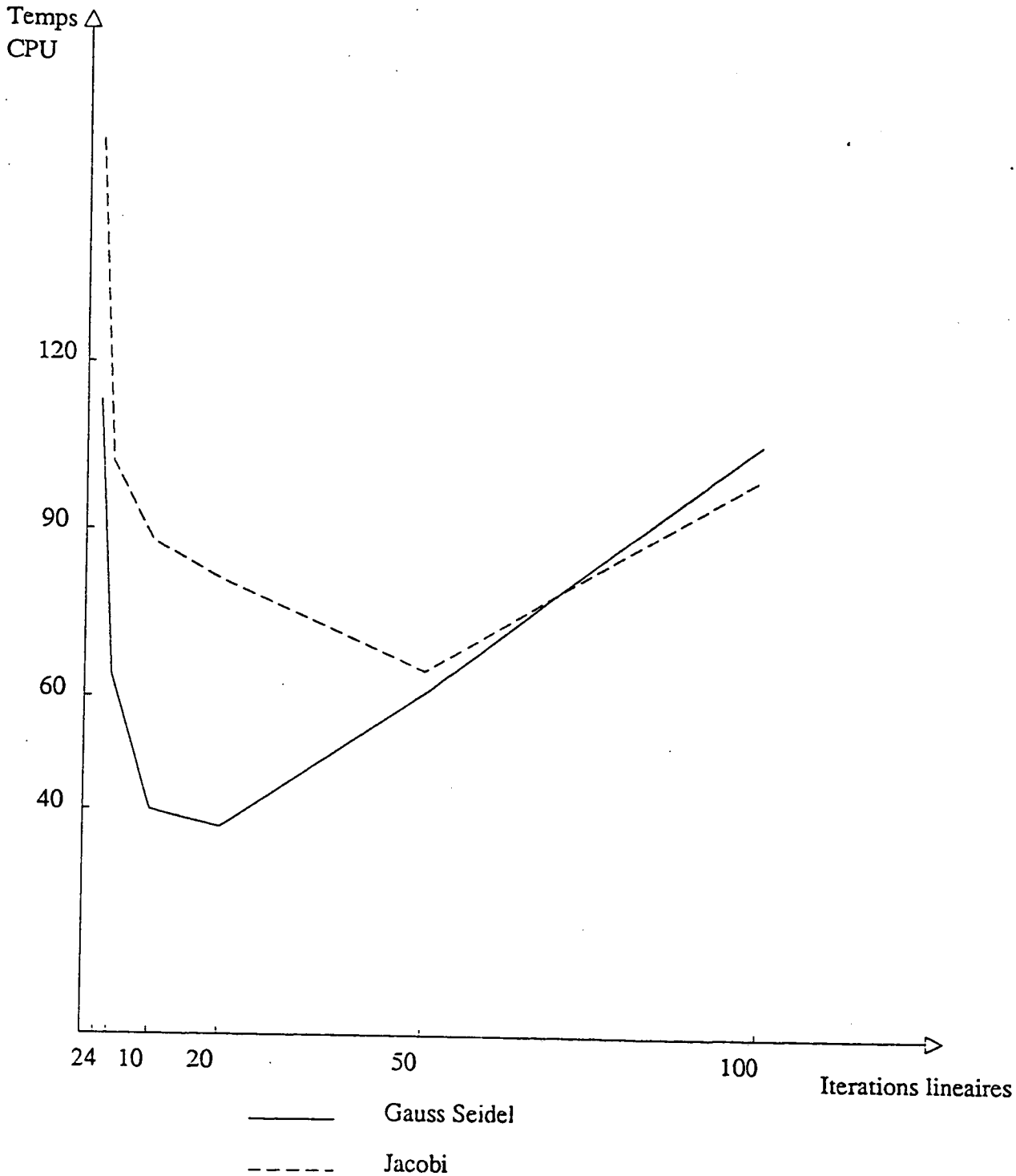
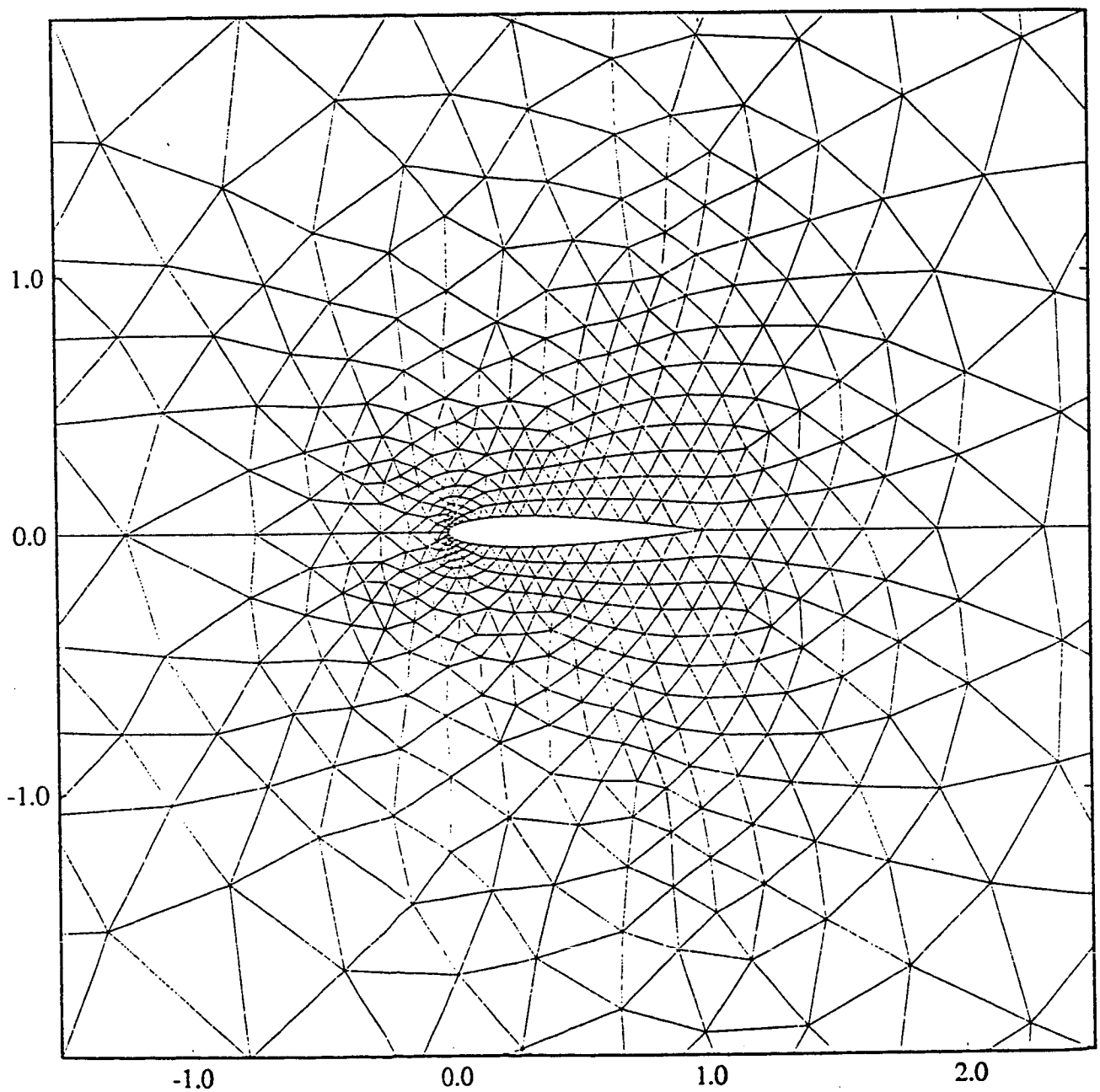


Figure 3 :
Géométrie : Profil d'aile d'avion NACA 0012
Nombre de points = 800



Courbe de convergence : NACA 0012
Implicite 50 Gauss-Seidel
CFL = fonction du Résidu

Figure 4.1: Ordre 1

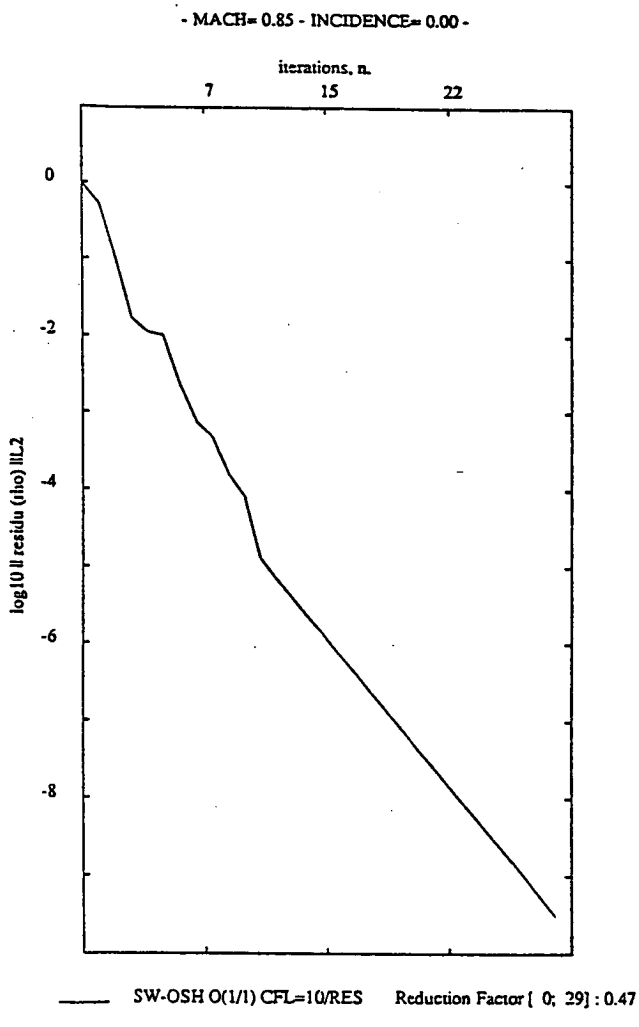


Figure 4.2 : Ordres 1 et 2

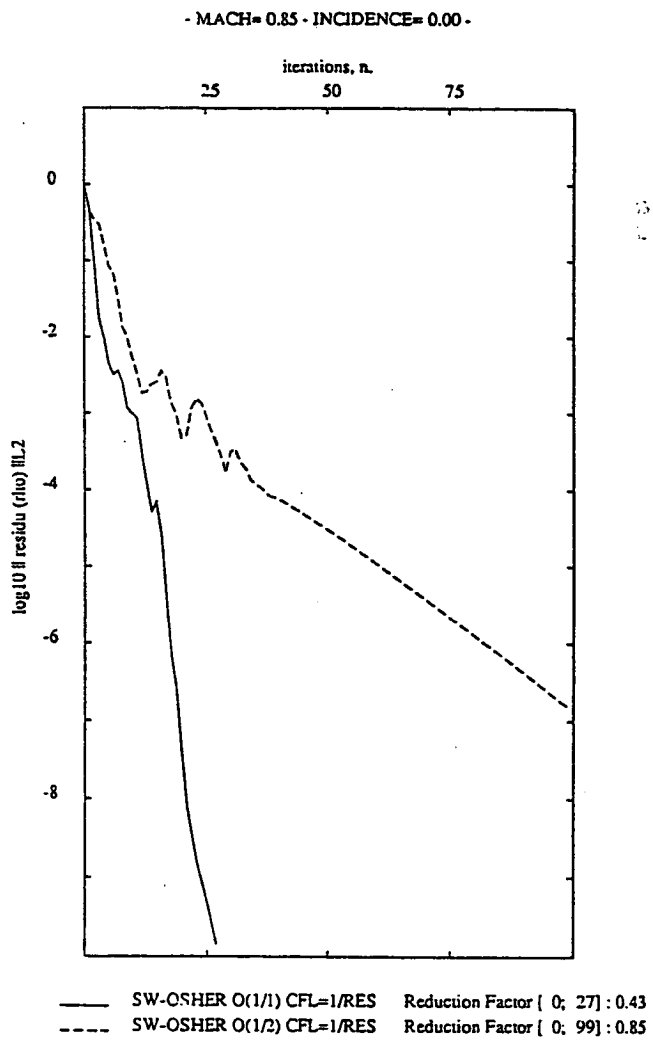
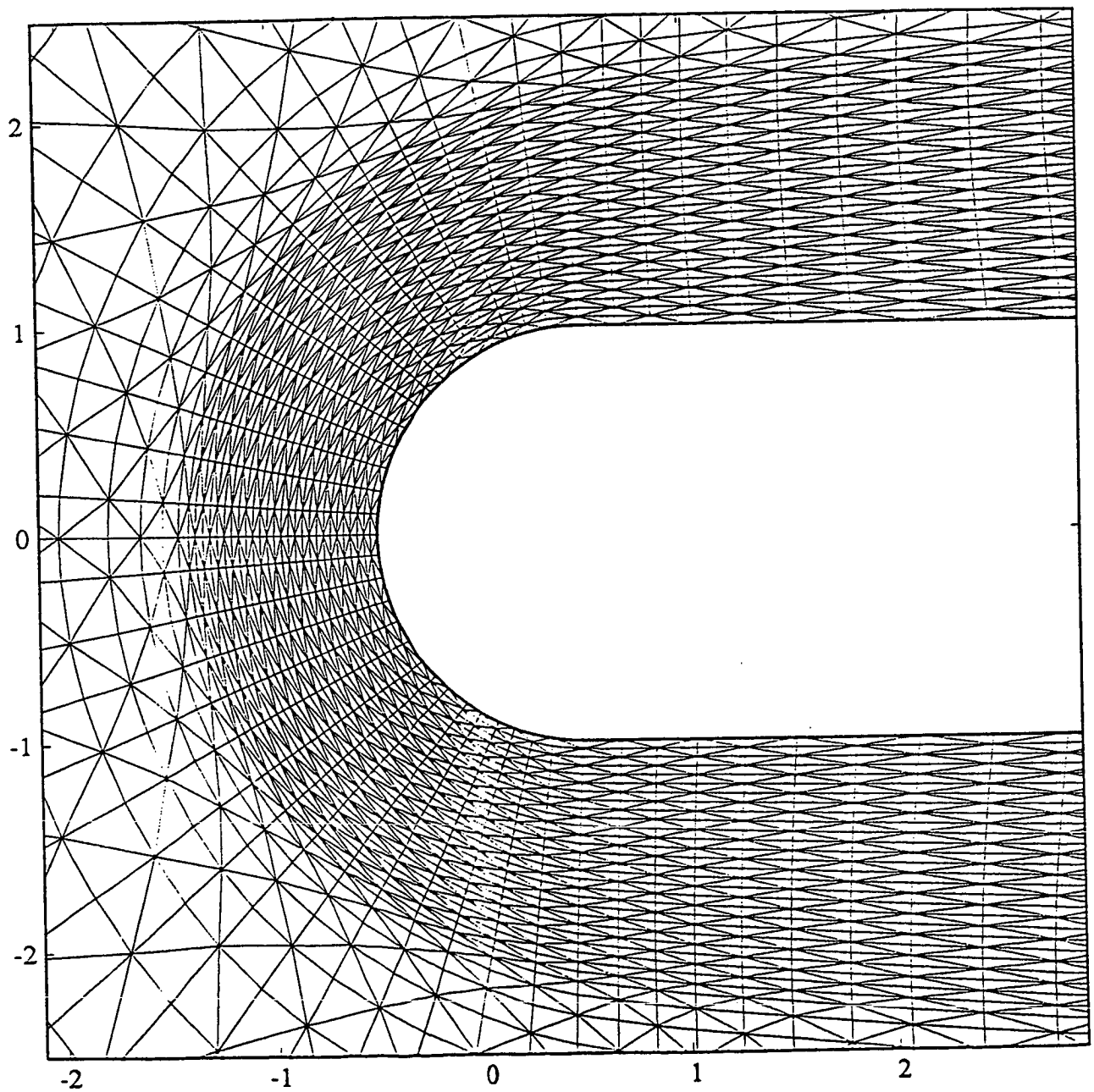


Figure 5.1 :
Géométrie : Corps émoussé à l'avant
Nombre de points = 1908

AGRANDISSEMENT DU MAILLAGE PRES DU CORPS ARRONDI



Solutions : Corps émoussé sans incidence
Mach = 8

Figure 5.2 : ordre 1

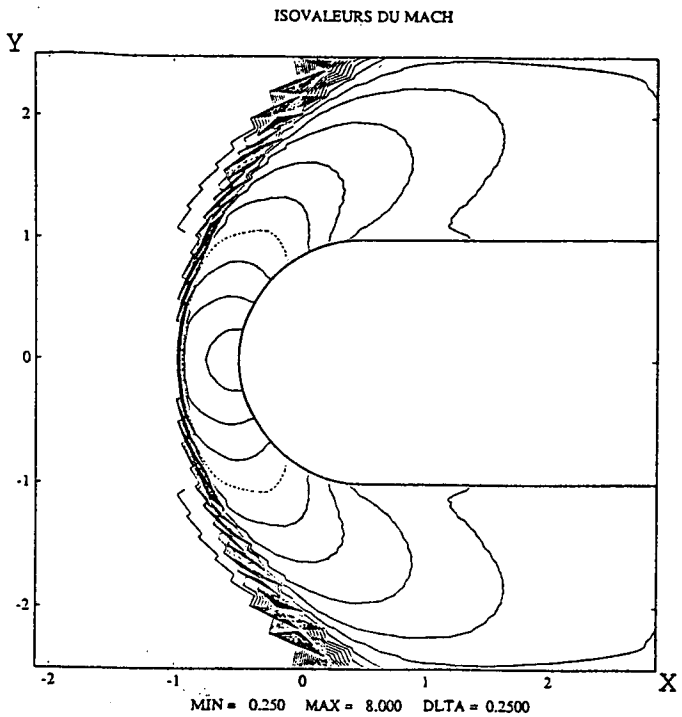


Figure 5.3 : ordre 1

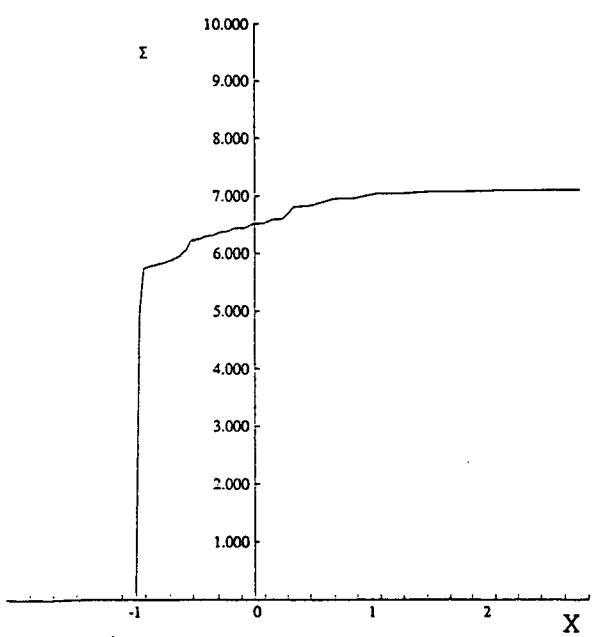


Figure 5.4 : ordre 2

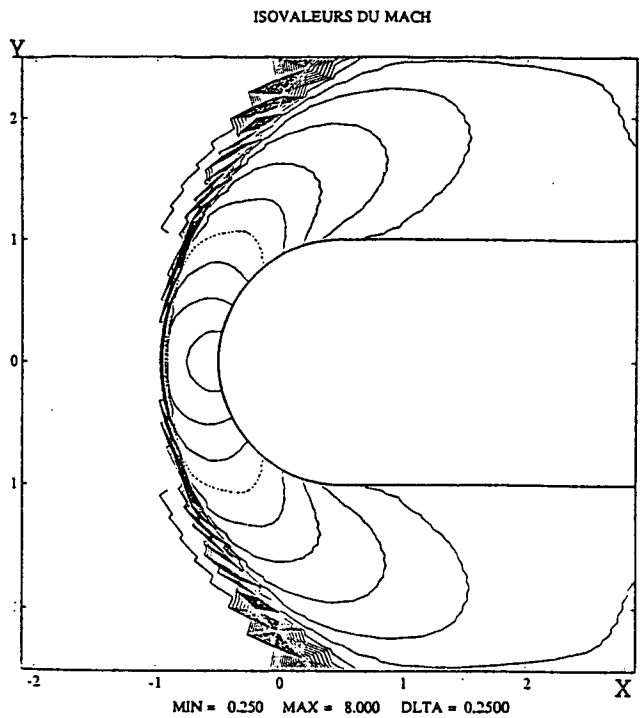
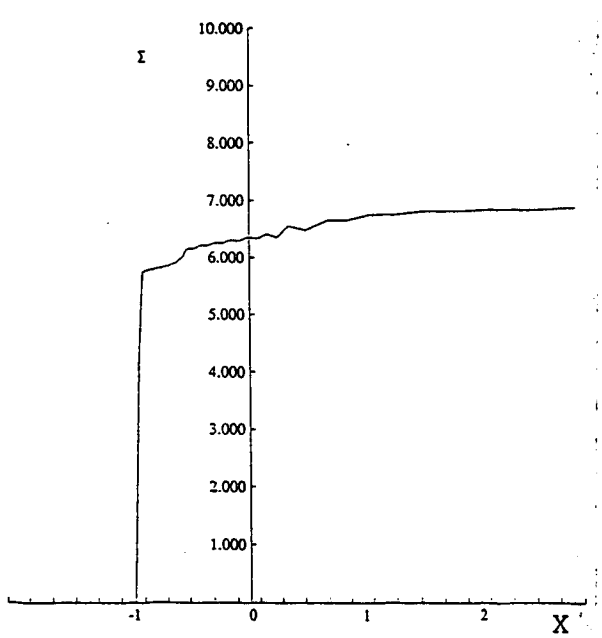


Figure 5.5 : ordre 2



Déviation de l'entropie :
si x ≥ 0.5 : Y = 0
si X > 0.5 : le long du corps (Y>0)

Solutions : Corps émoussé. Incidence = 30 degré
Mach = 8 - Ordre 2

Figure 5.6 :

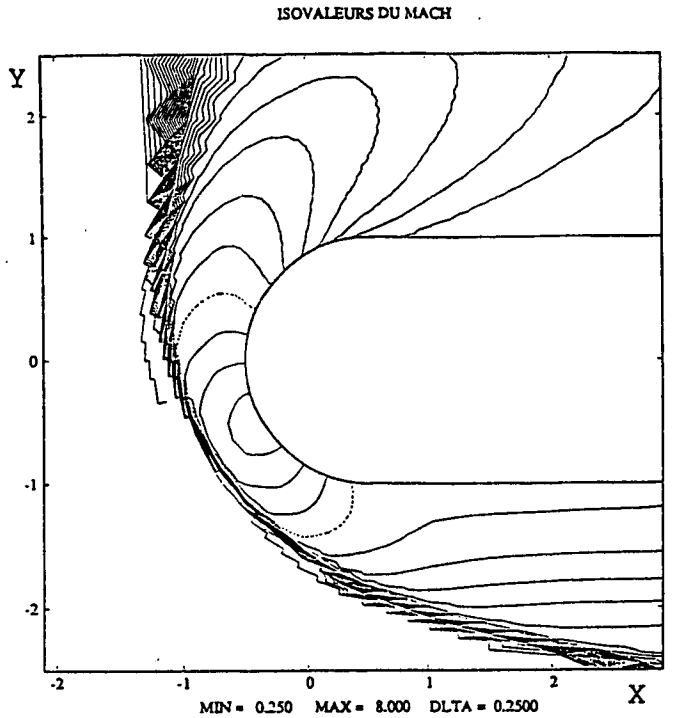
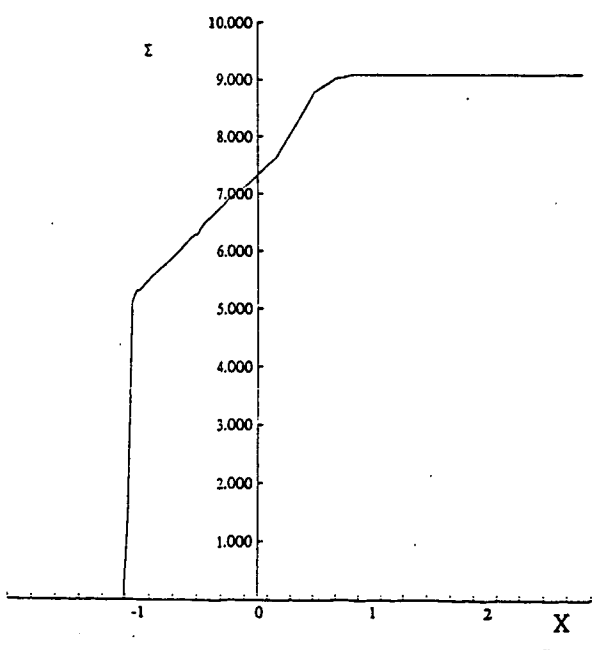


Figure 5.7 :



Déviation de l'entropie

Courbes de convergence : Corps émoussé - Mach = 8
CFL = $\frac{n}{2} \leq 30$

Figure 6.1 : ordre 1

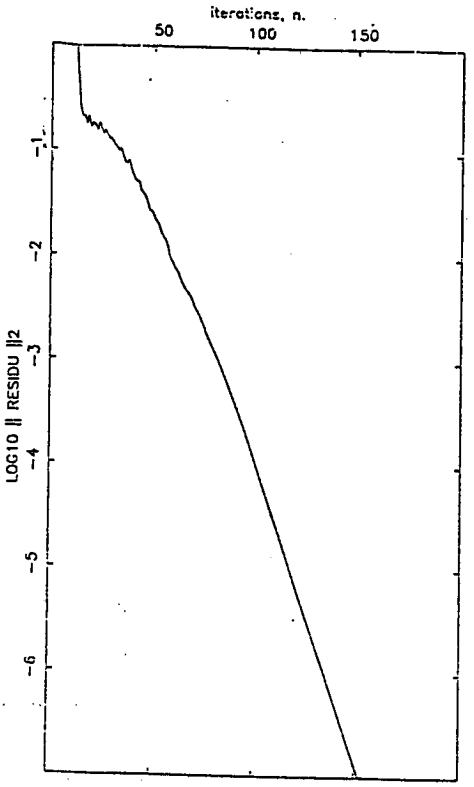


Figure 6.2 : ordre 2

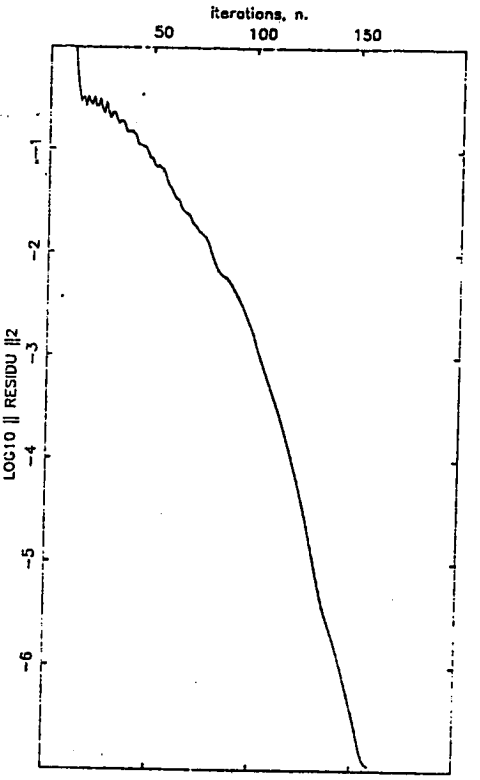


Figure 6.3 : ordre 2

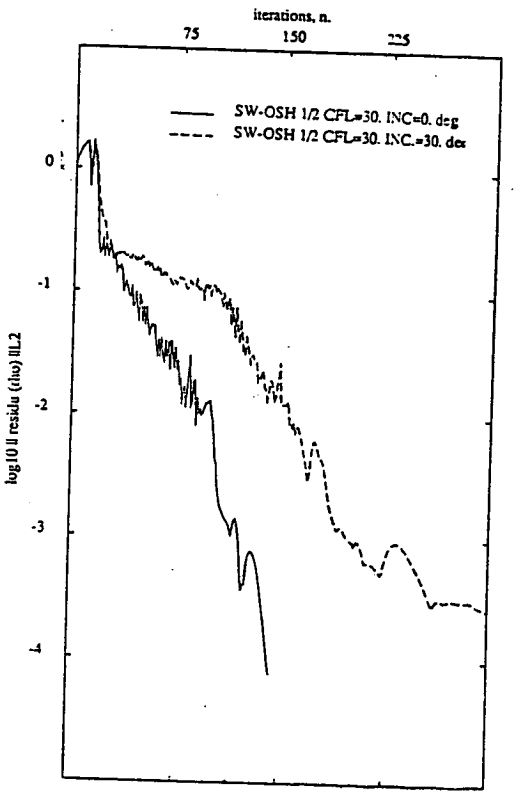


Figure 7 :

COUT DES SCHEMAS IMPLICITES LINEARISES ORDRE 1 ET 2
POUR UN CORPS ARRONDI 2-D (697 points)
HYPERSONIQUE (Mach infini = 8,00)

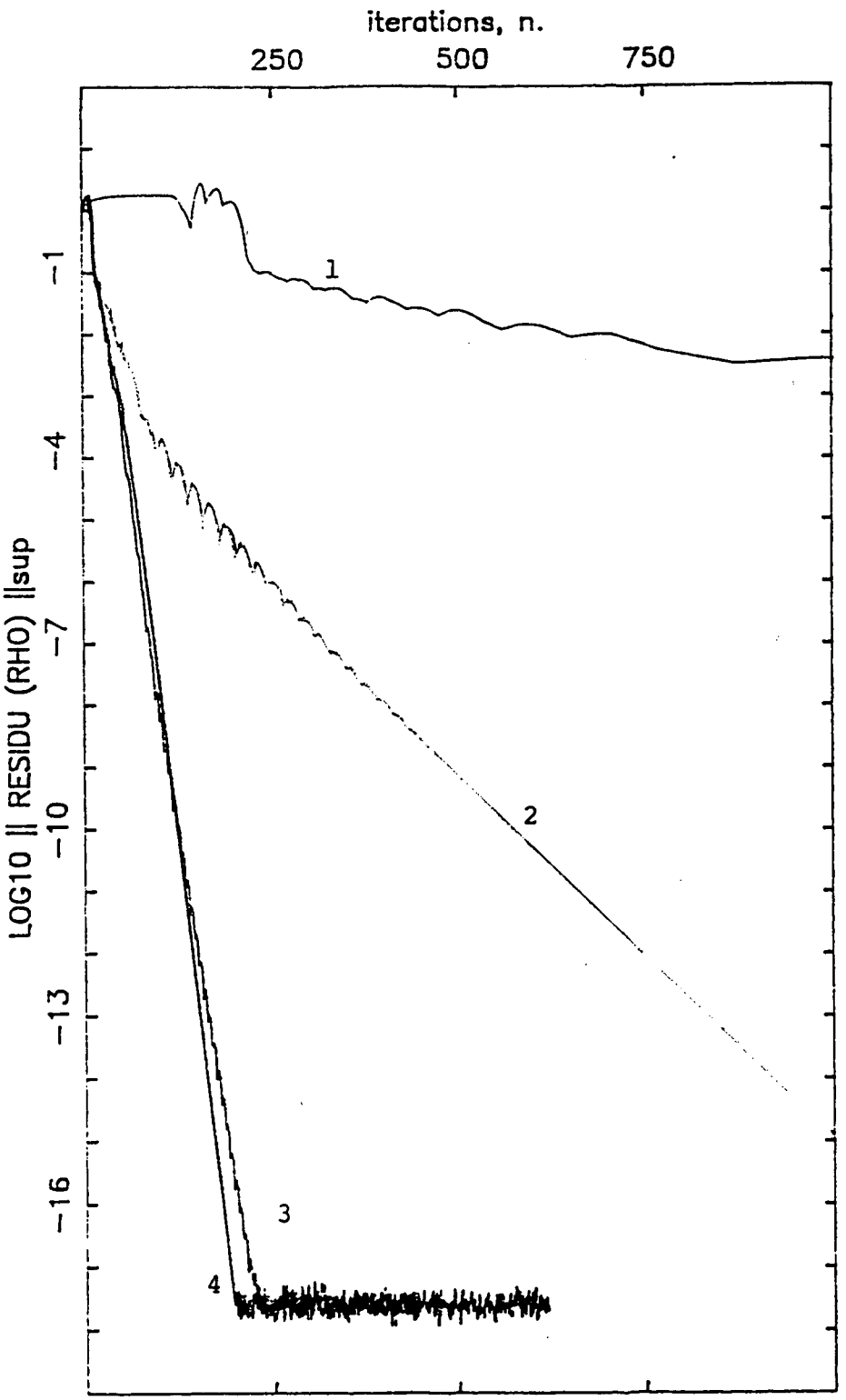
NATURE DU SCHEMA	COUT CPU (*) d'une iteration		COUT CPU (*) pour diviser le residu par 10000	EFFICACITE RELATIVE	Encombrement memoire
	Ordre				
EXPLICITE CFL = 0,6	1	12 sec	218 min	1	100 x NS
	2	16 sec	373 min	1	
IMPLICITE GAUSS-SEIDEL STOCKAGE MATRICIEL	1	54 sec	57 min	4	+ 112 x NS
	2	71 sec	106 min	3,5	
IMPLICITE JACOBI FAIBLE STOCKAGE MATRICIEL (**)	1	78 sec	76 min	3	+ 16 x NS
	2	104 sec	156 min	2,5	
PRECONDITIONNEUR DIAGONAL (**)	1	33 sec	58 min	4	+ 16 x NS
	2	44 sec	257 min	1,5	

(*) Temps machine sur DPS-68 (INRIA Sophia-Antipolis)

(**) Stockage de la diagonale uniquement

Figure 8.1 :

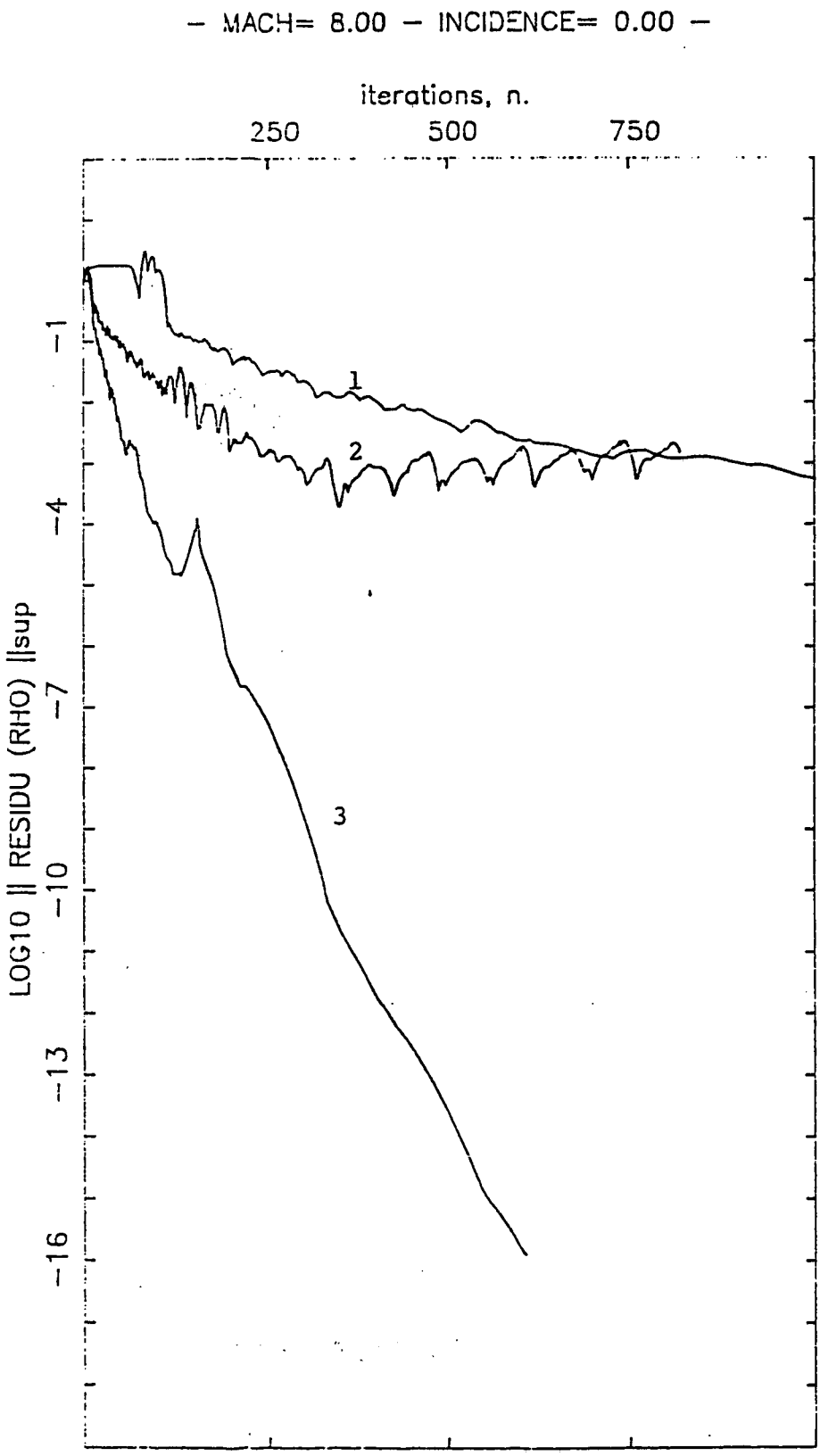
Courbes de convergence : Corps émoussé - ordre 1



1 : Schéma explicite
2 : Schéma préconditionnement diagonal
3 : Schéma implicite sous stockage - 4 Jacobi
4 : Schéma implicite avec stockage - 2 Gauss-Seidel

Figure 8.2 :

Courbes de convergence : Corps émoussé - ordre 2



1 : Schéma explicite
2 : Schéma préconditionnement diagonal
3 : Schéma implicite

bibliographie

[1] DERVIEUX A. ,
Steady Euler simulations using unstructured meshes
Cours au Von Karman Institute Lectures Series 85-04 (1985). Publié dans Partial Differential Equations of Hyperbolic Type and Applications, G. Geymonat Edt., World Scientist pp. 34-105 (1985).

[2] STOUFFLET B. ,
Résolution numérique des équations d'Euler des fluides parfaits compressibles par des schémas implicites en éléments finis
Thèse troisième cycle, Paris 6 (1984)

[3] VAN LEER B. ,
Towards the ultimate conservative difference scheme I. The quest of monotonicity
Lecture notes in Physics, Vol. 18 page 163 (1972)

[4] FEZOUI F. ,
Résolution des équations d'Euler par un schéma de Van Leer en éléments finis
Rapport de recherche $N°358$ INRIA Rocquencourt (1985)

[5] VIJAYASUNDARAM G. ,
Résolution numérique des équations d'Euler pour des écoulements transsoniques avec un schéma de Godunov en éléments finis
Thèse troisième cycle, Paris 6 (1982)

[6] STEGER J. - WARMING R.F.,
Flux vector splitting for the inviscid gas dynamic with applications to finite difference methods
Journal Comp. Physics, Vol. 40 $N°2$ pp 263-293 (1981)

[7] OSHER S. - SOLOMON F. ,
Upwind difference schemes for hyperbolic systems of conservation laws
Journal Math. Computation , (1982)

[8] ANGRAND F. - BOULARD V. - DERVIEUX A. - PERIAUX J. - VIJAYASUNDARAM G.,
Triangular finite element methods for the Euler equations
6th Internationnal conference on computing methods in applied sciences and engineering, Glowinsky R. et Lions J.L. edts, North holland (1984)

[9] DESIDERI J.A.,
Preliminary results on results on iterative convergence of a class of implicit schemes
Rapport de recherche $N°490$, INRIA Rocquencourt (1986)

[10] STOUFFLET B. - PERIAUX J. - FEZOUI F. - DERVIEUX A.,
Numerical simulation of 3-D hypersonic Eler flows around spaces vehicules
AIAA paper 87-0560 Reno, Nevada (1987)

[11] VAN LEER B.,
Computational methods for ideal compressible flow
Cours Von Karmann Institute, lectures series 1983-04 computational fluid dynamic (1983)

[12] EBERLE A. - MISEGADES K.,
Euler solution for a complete fighter aircraft at sub. and supersonic speed
58th AGARD Aix en Provence (1986)

[13] EBERLE A., SCHÄFER O.,
Hight order characteristic flux averaging for the solution of the Euler equations
Lecture notes in Methods of fluids mechanics, Vol. 13, Vieweg, Braunschweig-Wiesbaden (1986)

[14] ANGRAND F. - ERHEL J. ,
Vectorized finite element codes for compressible flows
Proc. of *Finite element in flow problem*, Antibes (1986). Wiley

[15] PEREZ E. - PERIAUX J. - ROSENBLUM J.P. - STOUFFLET B. - DERVIEUX A. - LALLEMAND M.H.,
Adaptative full multi-grid finite element methods for solving the two dimensionnal Euler equations
I.C. Numerical Methods for Fluid Dynamics Pekin/Beijing (1986) Springer

[16] LERAT A.,
Sur le calcul des solutions faibles des systèmes hyperboliques des lois de conservation à l'aide de schémas aux différences
Thèse , Paris VI (1981)

[17] LERAT A. - SIDES J. - DARU V.,
An implicit finite volume method for solving the Euler equations
In E.Krause (Ed), Eight Internationnal Conference on Numerical Methods in Fluid Dynamics (1982) Lecture notes in physics, 170, page 342-349 (1982)

[18] JESPERSEN C.D. - PULLIAM T.H.,
Flux vector splitting and approximate Newton methods
Computational Fluid Dynamics Conference AIAA. papier n°1899 page 535 (1983).

[19] FEZOUI F. - STEVE H.,
Décomposition d'un flux Van-Leer pour résoudre les équations d'Euler en éléments finis
Rapport de recherche INRIA Sophia Antipolis. A paraître.

[20] DESIDERI J.A. - HETTENA E.,
Numerical simulation of hypersonic equilibrium-air reactive flow
Rapport de recherche INRIA Sophia Antipolis. A paraître.

[21] VAN LEER B.,
Flux Vector Splitting for the Euler equations
Lecture Notes in Physics, vol. 170. page 405-512 (1982)

[22] ANDERSON W.K.- THOMAS J.L. - VAN LEER B.,
A comparison of Finite Volume Flux-Vector Splittings for the Euler Equations
AIAA paper 85-0122 (1985).

[23] B. VAN LEER - W. A. MULDER ,
Relaxation methods for hyperbolic conservation laws
Numerical Methods for the Euler Equation of Fluid Dynamics SIAM.
Edité par INRIA pp. 312-333 (1985) Philadelphia.

[24] D.J.A. WELSH - M.B. POWELL,
An upper bound for the chromatic number of a graph and its application to time tabling problems
Computer Journal, Vol. 10, $n°1$, pp. 85-86 (1967)

[25] R.F. WARMING - R.M. BEAM,
Implicit numerical methods for compressible Navier-Stockes and Euler equations
V.K.I., Lecture Series 1982-04, (1982)

[26] VARGA R.S.,
Matrix Iterative Analysis
Prentice-Hall. Serie dans Automatic Computation. G. Forsythe éditeur (1962)

[27] M-H. LALLEMAND - A. DERVIEUX,
A multigrid finite element method for solving the two-dimensional Euler equations
3rd Copper Moutain Conf. on Multigrids Methods, April 1987 Colorado. Multigrids Methods S. Mc Cormick ed. Marcel Dekker New-York.

[28] S. P. SPEKREIJSE - P. W. HEMKER,
Multigrid Solution of the steady Euler Equations
Notes on Numerical Fluid Mechanics, Volume 11, 33-44. Vieweg, Braunschweig. (1985)

[29] R.W. Mc CORMACK,
A Numerical Method for solving the Equation of Compressible Viscous Flows
AIAA Journal, 20 (1982).

[30] THOMAS J.L. - VAN LEER B. - WALTERS R.W.,
Implicit Flux Split for the Euler Equations
AIAA abstract, 19th Fluid Dyn. Cincinnati (1985).

[31] FEZOUI L. - STOUFFLET B.,
A class of implicit upwind schemes for Euler simulations with unstructured meshes
Rapport de Recherche INRIA No 517, Avril (1986).

[32] KRUZHKOV S.N.,
First order quasilinear equation with several independant variables
Traduction anglaise : Mat. Sbornik USSSR, vol 10, 217-243 (1970)

[33] LEROUX A.Y.,
Approximation de quelques problèmes hyperboliques non linéaires
Thèse d'état Rennes (1979).

[34] LAX P.D. ,
Weak solutions of Nonlinear Hyperbolic Equations and their Numerical Computation
Comm. on Pure and Appl. Math., 7 pp. 159-193 (1954).

[35] LAX P.D. - HARTEN A.- VAN LEER B. ,
On Upstream Differencing and Godunov Type Schemes for Hyperbolic Conservation Laws
SIAM Revue - Vol. 25 - No 1 (1983).

[36] SPEKREIJSE S.,
Multigrid Solution of the Steady Euler Equation
Thèse de doctorat. CWI Amsterdam (1987)

Remerciements :

Je tiens à remercier pricipalement Monsieur A. Dervieux, directeur de recherche, et Mlle L. Fezoui, chargé de recherche à l'INRIA Sophia- Antipolis, qui ont dirigé mon travail.

Je remercie amicalement toute l'équipe du projet SINUS de l'INRIA Sophia-Antipolis qui m'a suggéré des remarques, et m'a prodigué de nombreux conseils.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique